NR. INDEKSU
PL ISSN 0860–1674

7

Z MIKROKOMPUTEREM NA TY

Bajtek

MIESIĘCZNY DODATEK DO „SZTANDARU MŁODYCH" NR 7/86 CENA 100 ZŁ



100 TYSIĘCY

Piszę te słowa w momencie gdy kończy się X Zjazd Polskiej Zjednoczonej Partii Robotniczej. Wśród wielu spraw podnoszonych w czasie obrad ważkie miejsce zajęły również problemy i konsekwencje rewolucji informatycznej. Mówiono o nich podczas obrad plenarnych, zajmowano się nimi w zespołach problemowych, mocno zaznaczono w przyjętych przez X Zjazd dokumentach.

Sprostanie wyzwaniom rozwojowym, unowocześnienie kraju, nadanie właściwej rangi postępowi naukowo-technicznemu — te idee bliskie były wszystkim delegatom. Zostały one uwzględnione w założeniach społeczno-gospodarczego rozwoju kraju w latach 1986 i do roku 1995.

Jak stwierdził w swym wystąpieniu podczas X Zjazdu premier Zbigniew Messner: „Podstawowym czynnikiem unowocześnienia będzie elektronizacja, zwłaszcza przemysłu. W obecnym 5-leciu, w wyniku szerokiego stosowania mikroelektroniki w zespołach napędowych, automatycznego sterowania procesami spalania paliw, elektronizacji produkcji w hutnictwie, górnictwie, chemii nastąpi obniżenie zużycia energii. W niektórych dziedzinach sięgnie ono 15–20 proc. Rozpoczęta zostanie seryjna produkcja złożonych robotów i manipulatorów, elementów i podzespołów mikroelektroniki i hydrauliki siłowej. Nastąpi również około 40-proc. wzrost produkcji nowoczesnego sprzętu powszechnego użytku, w tym nowych odbiorników telewizyjnych i radiowych, magnetofonów, magnetowidów, laserowych odtwarzaczy płyt, a także mikrokomputerów. Poziom produkcji tych ostatnich osiągnie w roku 1990 około 100 tysięcy sztuk”.

Nikogo z tych, którzy naprawdę troszczą się o przyszłość naszego kraju nie trzeba przekonywać, co oznacza ta ostatnia podana przez premiera liczba. Doprowadzenie naszego przemysłu do takiego stanu aby już za 5 lat potrafił produkować 100 tysięcy sztuk komputerów rocznie — to stworzenie praktycznych, materialnych podwalin pod unowocześnienie kraju i pod powszechną edukację komputerową zarazem. Oczywiście, ta perspektywa powszechnej dostępności mikrokomputerów w całej ostrości stawia problem właściwego ich wykorzystania i oprogramowania. Zwłaszcza, że i zagrożeń w tym zakresie jest wiele. Sygnalizowano je otwarcie na X Zjeździe:

„Wydawałoby się, że mikrokomputery są ideologicznie obojętne — mówił Lesław Wojtasik, z-ca szefa GZP WP. — Mamy ich w tej chwili w Polsce około 150 tysięcy. Ilość ich rośnie lawinowo... Są urządzenia, ale nie wyprodukowaliśmy ani jednego programu, który by do nich pasował. Zostawiliśmy znowu wolne pole...”.

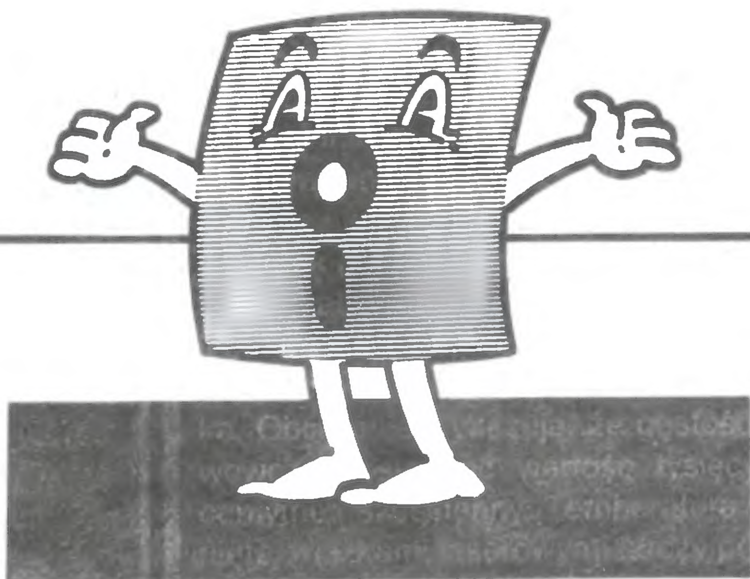
To samo zagrożenie miał na myśli Jerzy Majka, redaktor naczelny „Trybuny Ludu”, gdy mówił podczas dyskusji plenarnej: „Musimy jednocześnie wydać walkę przeciwnikowi w sferze techniki informowania. W produkcji wideofonów i wideokaset. W opracowaniu naszych programów dla gier komputerowych. W organizowaniu klubów wideo i komputerowych. W pełniejszym wykorzystaniu programów telewizji...”.

Że nie są to zagrożenia wydumane przekonywać nikogo chyba nie trzeba. Stawka idzie przecież o to, abyśmy włączając się do głównego nurtu światowej rewolucji naukowo-technicznej nie zatracili jednocześnie tych wartości kulturowo-światopoglądowych, które decydują o odrębności i tożsamości naszego narodu i państwa. Profesor Władysław M. Turski przestrzegał kilka lat temu, że jeśli nie weźmiemy się na serio za tworzenie rodzimej bazy programów to może nam wręcz grozić „anihilacja cywilizacyjna”. Wówczas obawy te uznano za przedwczesne. Dziś stały się one przedmiotem alarmu polityków!

W jaki sposób można wypełnić dramatyczną lukę braku polskich programów dla tego wszystkiego co nazywamy powszechnym ruchem komputerowym? Otóż odpowiedź jest trywialna: trzeba te programy przygotować. Trzeba opracować oparte na polskiej tradycji i kulturze gry komputerowe, trzeba podjąć gigantyczny wysiłek przygotowania różnorodnego oprogramowania dydaktycznego — z lekcjami historii, geografii i języka polskiego włącznie, trzeba „na polską modłę” przerobić pliki programów użytkowych, w pierwszej kolejności tych, które mogą być wykorzystywane w domach. Itp. itd... Wszystko oczywiście w języku polskim, z wykorzystaniem wszystkich liter polskiego alfabetu i zasad gramatycznych naszego języka.

To wszystko jest do zrobienia, tylko, żeby rzeczywiście zrobione zostało, trzeba wziąć się za robotę już teraz. Aby był dobry software potrzebne są dobre głowy i czas. Dobre głowy jeszcze w Polsce mamy, ale czas zaczyna być już czynnikiem krytycznym!

Waldemar Siwiński



WYBIERZ SAM

„BAJTEK” — MIESIĘCZNY DODATEK DO „SZTANDARU MŁODYCH”.
ADRES: 00-687 Warszawa, ul. Wspólna 61. Telefon 21-12-05.

ZESPÓŁ REDAKCYJNY: Waldemar Siwiński (z-ca redaktora naczelnego „SM” — kierownik zespołu), Oskar Bramski, Krzysztof Czernek, Wiesław Migut, Sławomir Polak, Roman Poznański, Wanda Roszkowska (opr. graficzne), Marcin Waligórski, Roman Wojciechowski.

Przewodniczący Rady Redakcyjnej: Jerzy Domański — redaktor naczelny „Sztandaru Młodych”.
WYDAWCA: RSW „Prasa-Książka-Ruch” Młodzieżowa Agencja Wydawnicza, al. Stanów Zjednoczonych 53, 04-028 Warszawa. Telefony: Centrala 13-20-40 do 49, Redakcja Reklamy 13-20-40 do 49 w. 403, 414.
Cena 100 zł.
Skład techniką CRT-200, przygotowalnia offsetowa i druk: PRASOWE ZAKŁADY GRAFICZNE RSW „PRASA-KSIĄŻKA-RUCH” w Ciechanowie, ul. Sienkiewicza 51.
Zam. nr 714/86, nakład 200.000 egz. P-101

OSTATNI DZWONEK



FOT. LEOPOLD DZIKOWSKI

Rozmowa z WŁODZIMIERZEM NATORFEM, nauczycielem fizyki w IX Liceum Ogólnokształcącym im. Klementyny Hoffmanowej w Warszawie, członkiem zespołu doradczego przy Ministrze Oświaty i Wychowania do spraw Edukacji Informatycznej.

— Od trzech lat prowadzi Pan w szkole mikrokomputerowe kółko zainteresowań. Jest Pan również członkiem zespołu doradczego. A jednak nie zdecydował się Pan na prowadzenie od nowego roku szkolnego przedmiotu — elementy informatyki. Czy uważa Pan, że jeszcze nie pora?

— Nic podobnego! Uważam wręcz, że jest ostatni dzwonek, żeby wprowadzić do szkolnej edukacji elementy informatyki. I chyba nie muszę uzasadniać tej opinii — zwłaszcza na łamach „Bajtka".

— Dysponuje Pan pracownią wyposażoną w cztery komputery MERITUM I oraz jeden ZX SPECTRUM+. Wprawdzie nierewelacyjnie — ale dla celów szkoły na początek wystarczy. Przez te trzy lata zdobył Pan także doświadczenie w pracy dydaktycznej...

— Zależy co rozumiemy przez pracę dydaktyczną. Jeśli chodzi o naukę programowania w wybranym języku, np. BASIC czy FORTRAN — to mam doświadczenie. Ale jeśli chodzi o nauczanie informatyki — to zdecydowanie nie! Natomiast w potocznej opinii uważa się, że programowanie i informatyka to jedno. Jeżeli jednak ustawić te dwa pojęcia we wzajemnej relacji — to programowanie jest tylko wycinkiem informatyki i sądzę, że wcale nie najistotniejszym. Jako fizyk muszę umieć programować, ale nie oznacza to, że znam wszystkie tajniki informatyki. Bywa, że układając program działam metodą prób i błędów, intuicyjnie. Moja wiedza informatyczna jest zbyt mała, żeby uczyć innych informatyki.

— Nie będzie większego błędu w stwierdzeniu, że jest Pan w podobnej sytuacji jak większość nauczycieli, którzy dziś prowadzą zajęcia z informatyki. Czy oznacza to, że nie mamy jeszcze warunków do wprowadzenia tego przedmiotu?

— Jesteśmy w takiej ciekawej sytuacji, że musimy wprowadzać elementy informatyki do szkół, czy nam się to podoba — czy nie. A jednocześnie nie są spełnione warunki do tego potrzebne. W tej chwili stoimy przed najważniejszym problemem: na jakich zasadach wprowadzać ten przedmiot? Wyobraźmy sobie, że w jednym państwie armia wyposażona jest w dzidy i łuki — w sąsiednim ma broń rakietową. Trzeba więc rozwiązać dylemat: czy armię zacofaną wyposażyć od razu w broń rakietową? Czy nie będzie to zbyt duży przeskok? Może na początek dać jej broń nieco mniej nowoczesną?

— Wyposażenie w nowoczesny sprzęt jest raczej niemożliwe, gdy weźmie się pod uwagę finansową zasobność oświaty...

— Dlatego pociągającą wydaje się propozycja pójścia na żywioł — jak ja to określam. Oczywiście — na początek. Obecnie mamy w nauczaniu infor-

matyki partyzantkę, dysponujemy — skromnym jeszcze — jakimś sprzętem i chętnym gronem słuchaczy. Na pierwszym etapie można bardzo wiele zdziałać metodami niezinstytucjonalizowanymi. Ale przyjdzie moment ujęcia tego w pewne ramy instytucjonalne: wprowadzenie podręcznika, programu, zaleceń metodycznych, wyposażenie w sprzęt i oprogramowanie, wyznaczenie języka programowania, który będzie wiodący itp. Zaletą takiej centralizacji jest to, że kończy się partyzantka, a zaczynają się regularne działania, które łatwo jest upowszechniać.

— Jakie są natomiast wady?

— Kończą się działania spontaniczne, a na obecnym etapie wyposażenia, wykształcenia nauczycieli itp. spełniają one ogromną rolę. Ale nie to jest najważniejsze — wprowadzenie informatyki na zasadach zajęć obowiązkowych (nawet do wyboru) może przynieść więcej szkody niż pożytku.

Młodzież garnie się do informatyki i to jest dla nauczyciela szalenie korzystne. W ramach informatyki są możliwości przemycenia olbrzymiej ilości wiedzy z innych przedmiotów. Podczas zajęć kółka zainteresowań siadają obok siebie uczniowie od klasy pierwszej po czwartą, o różnych zainteresowaniach. Uczniowie przychodzą na zajęcia z własnej, nie przymuszonej woli, nie są obciążeni perspektywą stawiania ocen, odpytywania. Traktują te zajęcia jako zabawę intelektualną, grę wyobraźni i logiki.

— Pojawia się więc pytanie, czy tej naturalnej ciekawości nie zabijemy perspektywą jeszcze jednego, szkolnego przedmiotu?

— Ponieważ uważam, że tak się stać może, staram się opóźnić wprowadzenie przedmiotu elementy informatyki w mojej szkole. Chyba, że ktoś by zdecydował, iż będzie to przedmiot bez klasówek, ocen... Ale to się z kolei kłóci z ogólnie przyjętymi zasadami: jak sprawdzać pracę ucznia, jak sprawdzać pracę nauczyciela itp.

— Te dylematy wiążą się z pracą całej szkoły, nie ograniczają się do informatyki.

— Tak, ale nie możemy zapominać, że takie przedmioty jak historia, fizyka, matematyka, biologia tkwią w tradycji i kulturze szkolnej. Uczniowie wiedzą czego mają się uczyć, czego mogą się na lekcji spodziewać. Tych przedmiotów uczyli się ich rodzice, mamy do czynienia z typowym powielaniem kultury — nawet, gdy programy się unowocześnia. Natomiast w przypadku wprowadzania nowego przedmiotu, stoimy u progu olbrzymich zmian kulturowych w szkole. Inaczej wprowadzanie informatyki będzie jedynie zabiegiem formalnym, który można będzie odnotować w sprawozdaniach i... nic się nie zmieni, nie wytworzy się nowych wartości.

— Jakie jeszcze inne bariery dostrzega Pan we wprowadzaniu nowego przedmiotu?

— Proponuję nie koncentrować się na sprawach sprzętu, hardware'u — każdy, choć trochę interesuje się tymi zagadnieniami, wie, jaka jest sytuacja. Po prostu nie mamy rodzimego komputera, który masowo można by zastosować w dydaktyce.

Inny problem to oprogramowanie. Nie ma co się łudzić, że powstaną programy wcześniej niż przedmiot — elementy informatyki. Dobry program dydaktyczny musi powstać w porozumieniu dobrego programisty z dobrym dydaktykiem. Dobrych programistów mamy, ale gorzej z dydaktykami, gdyż dobrym dydaktykiem może być tylko ten, kto wie czego będzie uczył. Nie ma zaś człowieka, który rozpoczynając naukę przedmiotu z góry wie, czego będzie uczył i jak to będzie robił. Dopiero po dwóch, trzech latach będziemy mogli stwierdzić, czy program, który ma być wprowadzony po wakacjach jest dobry. Dlatego między innymi tak ostrożnie podchodzę do nowego przedmiotu.

— A może Pańskie uprzedzenia wynikają z tego, że po prostu nauczyciele boją się tego nowego przedmiotu?

— Oczywiście, że się boją. Jest to zupełnie naturalne i nawet uważam to za pozytywne zjawisko — gdyby się nie bali, dopiero zaczęłaby się klęska w edukacji informatycznej. Ja mam pełną świadomość tego, że chłopcy, którzy zaczynali przychodzić na zajęcia, w prowadzonym przeze mnie kółku zainteresowań — dziś przerośli mnie o dwie, trzy głowy! Oczywiście — a może szkoda — było ich tylko kilku. Ale nie tego boi się nauczyciel — najdramatyczniejszą sytuacją w klasie jest moment, gdy odpowiadając na pytania uczniów nauczyciel dochodzi do granicy swojej wiedzy. Może jedynie wówczas odpowiedzieć: nie wiem. A w przypadku dopiero wprowadzonego przedmiotu, w sytuacji, gdy dopiero szkoli się nauczycieli — takich sytuacji może być wiele. Poza tym, żeby przekazywać wiedzę z jakiejś dziedziny — obojętnie, czy jest to historia czy matematyka, trzeba wiedzieć znacznie więcej niż przewiduje program nauczania, trzeba mieć tę wiedzę usystematyzowaną, swobodnie się w tej dziedzinie poruszać. A czy kilkumiesięczny kurs sprawę rozwiąże?

— Jak w takim razie ocenia Pan program przedmiotu elementy informatyki?

— Ja w ogóle nie jestem w stanie go skomentować! Na przykład program przewiduje nauczanie LOGO. Ja mogę to ocenić z punktu widzenia fizyka-użytkownika, a nie informatyka. Mogę stwierdzić, że LOGO w porównaniu z BASIC-em jest to przerost formy nad treścią, ale czy mój sąd będzie słuszny? Dziś nikt autorytatywnie nie może ocenić, na ile dobry jest to program. Zweryfikuje go praktyka. Jedyne, co mogę stwierdzić, to że masowo powinniśmy kształcić użytkowników informatyki, a tylko wąskie grono uczniów zostanie informatykami. Nie można popełnić błędu i na siłę kształcić samych informatyków. Około 70 proc. młodzieży będzie w przyszłości jedynie korzystać z informatyki. Ten przedmiot powinien nauczyć rozeznania co to jest informatyka i jak się nią posługiwać.

— Czy program stwarza takie możliwości?

— Nie jest to sprawa programu a podejścia nauczyciela. Dla rozsądnego nauczyciela program jest jedynie wskazówką. O realizacji programu muszą decydować warunki konkretnej szkoły: zainteresowania uczniów, posiadany sprzęt — a wcale nie jest powiedziane, że każda szkoła powinna być jednolicie wyposażona w komputery. Stosowanie kilku typów sprzętu jest korzystniejsze z punktu widzenia dydaktyki. I tu wracamy do mojego przekonania, że nie chcę jeszcze w tym roku wprowadzać nowego przedmiotu. Za dużo jest jeszcze niewiadomych, niewielkie jest także przygotowanie, zaplecze itp. Oczywiście w innych szkołach i w odczuciu innych nauczycieli może być inaczej. Niemniej — ostrożność — nie zaszkodzi.

Rozmawiali:
Sławomir Polak
Roman Wojciechowski

KOMPUTERY W RENESANSIE

Co ma komputer do renesansu? — Może niejeden się z Was zapytać. Okazuje się, że ma. Ba, nawet bardzo do twarzy komputerowi w renesansowym otoczeniu. Mogli się o tym przekonać mieszkańcy Zamościa, w pierwszą niedzielę czerwca.

W podcieniach Rynku Wielkiego rozsiadły się mikrokomputery. Dla początkujących były atrakcyjne gry, dla zaawansowanych — zadania z zakresu programowania oraz wielki turniej „Komputer bez tajemnic". Do eliminacji pisemnych stanęło 120 uczniów zamojskich szkół. Najlepszych sześciu wzięło udział w finale. Zwycięzcą okazał się Grzegorz Teresiński z I LO im. Jana Zamoyskiego.

— Dzięki „Bajtkowi" i „SM" wiele osób zobaczyło po raz pierwszy w akcji mikrokomputery. Mogli spróbować owych sił nie tylko w grach, ale i w programowaniu. I co najważniejsze, przekonać się, że nie święci garnki lepią.

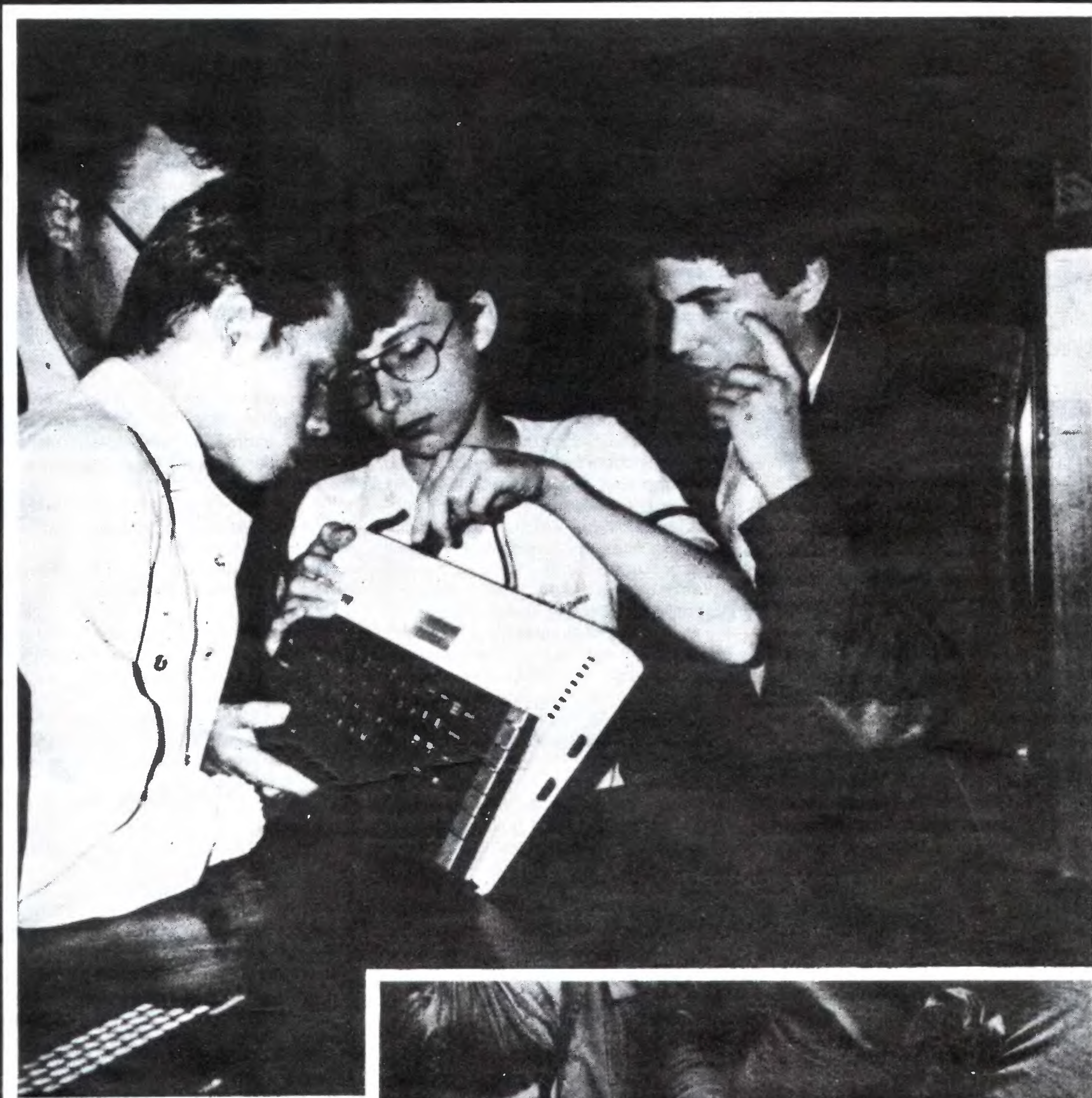
Dlaczego właśnie „Bajtek" pojechał do Zamościa? Po pierwsze, gdyż zaprosił nas Wojewódzki Zarząd ZSMP wraz z Młodzieżową Agencją Kulturalną a po wtóre działa tam bardzo prężnie pod patronatem I LO klub mikrokomputerowy, mający ambicje przyciągnąć młodzież, interesującą się tą problematyką. W dniu naszego przyjazdu klub ten na uroczystym spotkaniu przyjął oficjalną nazwę „Bajtek": jako pierwszy na Zamojszczyźnie.

Nasze liceum **— mówi mgr inż. Wiesław Włoszczyński —** ***jest pierwszym w regionie, w którego programie zajęć pozalekcyjnych znalazło się miejsce dla mikrokomputerów. Chętnych mamy wielu, ale barierą jest sprzęt. Dlatego irytuje mnie fakt, że choć inne szkoły na terenie naszego województwa również go dostały, to nikt z niego nie korzysta.***

„Bajtek" oczywiście nie pojechał z pustą ręką. Skromny stan posiadania zamojskiego klubu powiększył o mikrokomputer „Atari 800 XL".

W naszej szkole **— powiedział dyr. I LO Bolesław Hass —** ***ten drogi sprzęt nie będzie zamknięty na cztery spusty. Od 1 września w liceach ogólnokształcących zostaną wprowadzone przedmioty uzupełniające. Jedną z nich ma być informatyka. W naszej szkole będzie to możliwe, jeśli będziemy mieć co najmniej pięć mikro-***





komputerów. Oczywiście nie jesteśmy sobkami i każdy młody człowiek, którego interesuje ta nowoczesna technika ma u nas otwarte drzwi. Myślimy też o komputerach nie tylko jako o zabawie. Chcemy, aby miały praktyczne zastosowanie m.in. w naszej bibliotece, pomagały prowadzić lekcje z matematyki, fizyki, chemii czy biologii.

Komputerowego szaleństwa nie da się już zatrzymać — twierdzi dyrektor zamojskiego MAK-u, Leonard Marczuk. Będziemy wszystko robić, aby rozwijać kluby mikrokomputerowe na terenie naszego województwa, choć zdaję sobie doskonale sprawę, że największym hamulcem jest sprzęt. Podobno najtrudniejszym jest pierwszy krok, ale my już go zrobiliśmy.

Oskar M. Bramski





FOT. JACEK BARCZ

MIKROPROCESOR

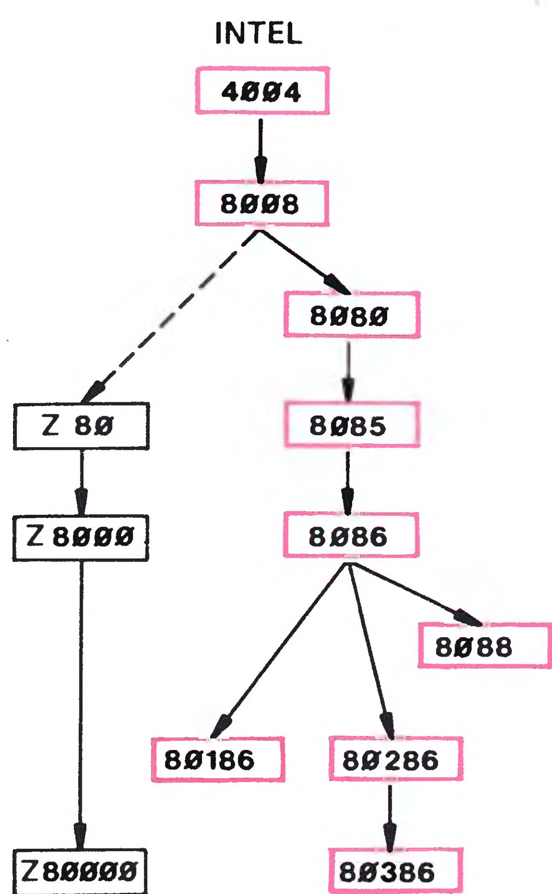
WCZORAJ...

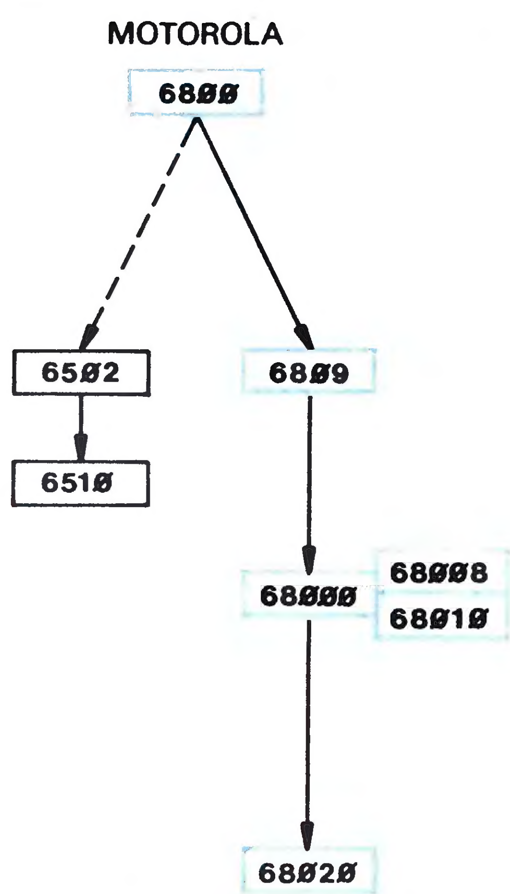
Pojawił się w laboratoriach przez przypadek. Japońska firma E.T.I. produkująca drogie kalkulatory biurowe zamówiła w firmie Intel nowy rodzaj układu elektronicznego do swoich wyrobów. Marcin „Ted" Hoff z Intel'a realizując ten projekt, postanowił uzupełnić funkcje układu o pewne możliwości programowania. Tak powstał w 1971 r. Intel 4004 — mikroprocesor 4-bitowy.

W roku 1973 powstał układ **8008** — pierwszy mikroprocesor 8-bitowy. Opracowany przez **Intel** dla firmy **Computer Terminal Corp.** obecnie **Datapoint**, procesor ten miał jednak jeszcze bardzo prymitywną architekturę i zasługiwał raczej na miano sterownika, niż mikroprocesora ogólnego zastosowania (**MPU**-microprocessor unit).

Następny w kolejności — **Intel 8080** — mimo, że strukturalnie jest potomkiem układu **8008**, to jednak z uwagi na zestaw instrukcji — był już prawdziwym MPU. Mikroprocesor ten został mózgiem pierwszego komercyjnego mikrokomputera, ale to zupełnie inna historia.

MIKROPROCESOR





Przyjrzyjmy się bliżej mikroprocesorowi 8-bitowemu na podstawie **Intela 8Ø8Ø** i jego otoczenia.

Pamięć 8-bitowego procesora jest zorganizowana w komórki (bajty), z których każda zawiera 8-bitów (stanów zer lub jedynek) informacji. Jeden bajt może zawierać kod litery, liczby lub może reprezentować jedno słowo instrukcji w kodzie wewnętrznym mikroprocesora (w języku maszynowym).

Procesor 8-bitowy posiada 16-bitową „szynę adresową". Oznacza to, że może on odczytać lub zapisać 2^{16} czyli 65.536 bajtów. Kolejnym komórkom pamięci przyporządkowane są adresy od Ø do 65.536, będące przestrzenią adresową procesora 8-bitowego. Często liczy się duże ilości pamięci jako wielokrotność liczby 2^{10} = 1Ø24, co w przybliżeniu jest równe 1ØØØ i oznacza się popularnym skrótem K (kilo). Stąd też pochodzi magiczna liczba ograniczająca nam ilość bajtów pamięci MPU 8-bitowego do 64kB (65.536 bajtów).

MPU zapisuje i odczytuje bajty z pamięci pojedynczo, jeden po drugim. Podczas pojedynczego cyklu „pobrania zawartości" pamięci przemieszczane jest zatem tylko 8 bitów. **MPU** 8-bitowy ma instrukcje działające na 16 bitowych liczbach, jednakże robi to w dwóch podejściach po jednym bajcie. Niezależnie od programu, którym posługuje się programista, procesor wykonuje zawsze ciąg instrukcji w kodzie maszynowym. Pojedynczo tworzą one raczej prymitywne czynności, takie jak przemieszczanie bajtu między określonym adresem w pamięci, a „rejestrem" **MPU** lub dodanie do siebie dwóch 8 bitowych liczb, które mają być pobierane z pamięci. Program taki w postaci kodu maszynowego reprezentowany jest poprzez sekwencje bajtów zapamiętanych kolejno w pamięci. Niektóre instrukcje wymagają tylko jednego bajtu, większość od dwóch do czterech (np. w procesorze Z-8Ø) zapamiętanych zawsze w następujących po sobie bajtach.

DZIŚ

Większość dużych, znanych firm komputerowych nie może jednak pozwolić sobie na to, by nie wejść na rynek ze sprzętem 16-bitowym. Pierwszą firmą produkującą w 1974 r. 16-bitowy procesor była National Semiconductor. W 1978 r. zjawił się Intel 8Ø86, w latach kolejnych 8Ø88 (16-stka z 8-bitową szyną danych), 8Ø186 i 8Ø286. W 1979 r. firma Zilog wyprodukowała Z8ØØØ, w 198Ø na rynek trafił procesor Motoroli MC68ØØØ, a w 1982 MC68Ø1Ø.

Rynek mikrokomputerowy podzielił się dziś na dwie części: Z jednej strony mamy do czynienia z serią komputerów osobistych firmy IBM (PC, XT, AT) i całą plejadą firm produkujących komputery kompatabilne (całkowicie odpowiednie) z maszynami IBM, takie jak AT T, ITT, Kaypro itd., bazujących na procesorach Intel 8Ø86, 8Ø88, 8Ø186 i 8Ø286. Drugą część tworzą komputery na bazie rodziny procesorów Motoroli MC68ØØØ, produkowane przez takie firmy jak Sinclair (QL), Commodore (Amiga), Atari (ST), Apple (Macintosh, Lisa), Hewlett-Packard (HP-IPC). Ceny większości tych komputerów są już porównywalne z wczorajszymi cenami komputerów 8-bitowych i wahają się w granicach 25Ø $ — 700 $.

W procesory inne, niż wymienione, wyposażony jest jedynie niewielki procent komputerów.

Jak porównywać mikroprocesory? Wydaje się to łatwe. 16-bitowy jest dwukrotnie potężniejszy, niż 8-bitowy, a 32-bitowy — czterokrotnie. Ale czy tak jest rzeczywiście?

Procesor 8-bitowy może posiadać tylko 2^8 = 256 różnych instrukcji. Liczba ta może być zwiększana, ale są to już trochę sztuczne sposoby. MPU 16-bitowy może mieć aż 2^{16} = 65.536 niezależnych instrukcji. Jest to na pewno liczba o wiele za duża na potrzeby programowania. Zupełnie wystarczające wydaje się przeznaczenie 9 bitów na instrukcje i pozostawienie 7 na dane (wszystko w jednym słowie i w jednym cyklu pobrania pamięci).

Następną cechą jest liczba wewnętrznych rejestrów. W porównaniu z procesorami 8-bitowymi liczba ich jest oszałamiająca np. Motorola MC8ØØØ posiada 17 rejestrów z czego 15 32-bitowych. Nie ma rejestrów specjalnie wyróżnionych, (będących odpowiednikiem akumulatora). Większość instrukcji może być wykonywanych w każdym rejestrze, zatem nie trzeba już przekazywać wszystkich czynności obliczeniowych procesora przez jedyny akumulator i „żonglować" rejestrami. Nikogo nie trzeba chyba przekonywać, jak bardzo upraszcza to programowanie.

Zwiększona liczba instrukcji obejmuje takie rozkazy, jak mnożenie i dzielenie. Funkcje te, jak pewnie niektórzy wiedzą, nie są takimi trywialnymi procedurami pisanymi w języku maszynowym MPU 8-bitowego. Procesor 16-bitowy może również wykonywać te rozkazy w sposób programowy (rekursywny), a nie „hardware'owy", ale programista już nie musi, a nawet nie wie o tym.

Dodano również wiele nowych spsobów adresowania pamięci, co umożliwiło implementację, na 16-bitowe mikrokomputery takich wielozadaniowych systemów jak UNIX i spowodowało duży postęp jakościowy.

Przyjrzyjmy się temu bliżej. Procesory 8Ø86, 8Ø88, MC68ØØØ posiadają 2Ø bitowe szyny adresowe, co daje 2^{20} = 1.Ø48.576 bajtów (1 MB) bezpośrednio adresowalnej pamięci. Procesor MC68ØØØ posiada nawet 24-bitową szynę adresową co umożliwia adresowanie 16MB.

Przy omawianiu MPU 16-bitowego należy wspomnieć o „pamięci wirtualnej" i „maszynach wirtualnych". Procesorem o tych możliwościach jest np. MC68Ø1Ø Motoroli. Koncepcja pamięci wirtualnej polega na tym, żeby dać programiście poczucie poruszania się w niemal nieograniczonej przestrzeni adresowej, ewentualnie stworzenia możliwości emulacji (imitacji) urządzeń peryferyjnych (dysków, drukarek, monitorów itp.) nie występujących fizycznie w systemie (to nazywa się właśnie maszyną wirtualną). Podstawowym mechanizmem pamięci wirtualnej jest zapewnienie procesorowi tylko pewnej ograniczonej ilości szybkiej pamięci fizycznej (operacyjnej), która może być adresowana bezpośrednio przez komputer w czasie gdy trzymamy obraz całej, o wiele większej wirtualnej pamięci, na wtórnym nośniku informacji np. dyskach o olbrzymiej pojemności. W chwili gdy procesor ma zamiar pobrać lub zapisać komórkę o adresie poza aktualnie dostępną przestrzenią adresową, wykonanie programu zostaje na chwilę automatycznie zatrzymane, dana część informacji aktualnie potrzebnej zostaje przepisana szybko do pamięci operacyjnej i program wykonuje się dalej. Procesor MC68Ø1Ø posiada tę wyjątkową cechę, że instrukcja wymagająca zmiany pamięci jest na czas tej zmiany zapamiętana i przerwana, a po zmianie odtworzona i kontynuowana dalej, w odróżnieniu od innych metod gdy powtórzenia wymagała cała powyższa instrukcja od początku.

Inną cechą MPU 16-bitowego jest umożliwienie współbieżności wykonywania pewnych operacji w mikroprocesorze. Normalnie przyzwyczajeni jesteśmy do szeregowego sposobu pobierania informacji z pamięci i realizowania kolejnych instrukcji. Nie jest to jedyne rozwiązanie. Niektóre operacje można wykonywać równolegle. Np. w procesorze 8Ø88 wydzielono tzw. układ sprzężenia z magistralą. Jest on odpowiedzialny za dostarczenie bajtów z pamięci i umieszczanie ich w kolejce w buforze procesora, podczas gdy układ centralny wykonuje w kolejności operacje na tych bajtach i nie zajmuje się już wypracowaniem adresów i cyklami pobrania tych słów z pamięci.

Na końcu należy wspomnieć o większej częstotliwości zegara z jaką może pracować MPU 16-bitowy. Typowymi częstotliwościami pracy są 7–12 MHz, a częstotliwości maksymalne dochodzą do 14 MHz.

Możemy już chyba podjąć próbę odpowiedzi na pytanie zadane na początku. Jak porównywać procesory? Są cztery podstawowe cechy, które należy wziąć pod uwagę. Są to:

1. Prędkość: maksymalna częstotliwość zegara z jaką może pracować MPU.
2. Przestrzeń adresowa: maksymalny obszar pamięci jaki MPU może bezpośrednio adresować.
3. Lista instrukcji: hasło to obejmuje ilość i jakość „garnituru" instrukcji MPU.
4. Szerokość słowa: ilość bitów, które MPU może przetworzyć za jednym razem.

Jak widać nie jest to jednoznaczna odpowiedź i nie można tych cech ułożyć w jakąś konkretną hierarchię, chociaż szerokość słowa odgrywa istotną rolę, gdyż pozostałe trzy parametry są mocno związane z tą cechą.

JUTRO

Nie ulega wątpliwości, iż w najbliższych latach ruszy produkcja mikrokomputerów 32-bitowych. Właściwie już się to dzieje w przypadku komputerów osobistych. Można się spytać, po co? Odpowiedzi jest wiele: większa prędkość, wielozadaniowość, wielodostępność, kompatybilność z dużymi komputerami, możliwość wykonywania olbrzymich programów np.: „sztucznej inteligencji", systemów obróbki danych, rozpoznawania mowy i chyba obecnie najważniejsza możliwość wprowadzenia do mikrokomputerów systemu operacyjnego UNIX.

I znów jako pierwsza na rynku pojawiła się firma Intel z procesorem 8Ø386, który być może znajdzie swoje miejsce w kolejnych generacjach IBM-PC. Drugą firmą jest Motorola, która uruchomiła produkcję procesora MC 68020, który połączył w jedno 32-bitowe możliwości z pełną kompatybilnością z całą już bardzo popularną rodziną MC 68ØØØ.

Na rynku znajdują się też już kości innych firm: Zilog (Z8ØØØØ, NCR 32ØØØ), Hewlett-Packard, Inmos, Fairchaild Camera, National Semiconductor, Texas Instruments, Western Electric itd.

Kto wygra rywalizację? Na pewno pozostaną Intel i Motorola. Z pozostałych może National Semiconductor ze swoim procesorem 32ØØØ (możliwość bezpośredniego adresowania 3ØØMB). Na pewno też trzeba wziąć pod uwagę Japonię. Firma NEC ogłosiła, że pracuje nad 32-bitowym procesorem z 7ØØ.ØØØ tranzystorów w układzie. Termin ukończenia prac — 1987 r. Firma Hitachi potwierdziła zakończenie prac wstępnych nad 32-bitowym procesorem. Termin rozpoczęcia masowej produkcji — 1986 r. Właśnie firmy japońskie mogą być „czarnym koniem" wyścigu.

Czym się te układy charakteryzują lub będą charakteryzowały? Typowe jest dalsze scalanie elementów komputera w jednym układzie np.: RAM i ROM w kości, powiększanie liczby wewnętrznych rejestrów, listy rozkazów, ilości pamięci bezpośrednio adresowalnej, jak i poszerzanie wirtualnych możliwości procesora oraz zwiększanie szybkości zegarów.

A co dalej? 64 bity, 128 itd? Czy skazani będziemy na ciągłe śledzenie tej gonitwy pozornie bez konca?

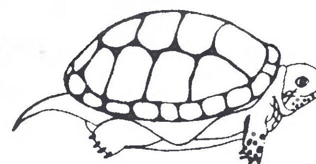
Procesor 64-bitowy spodziewany jest pod koniec lat osiemdziesiątych. Sądzę, że spotka go jednak znacznie mniejsze zainteresowanie, niż procesory 32-bitowe. Większość dużych komputerów obecnie używa 32-bitowych słów i niecelowe wydaje się zwiększanie tej wartości. Najważniejszy cel to doprowadzenie do całkowitej kompatybilności dużych i małych komputerów. Zostanie to osiągnięte przez wprowadzenie 32-bitowego MPU.

Niewątpliwie na uwagę zasługują też procesory typu RISC (Reduced Instruction Set Computers) o zmniejszonej liczbie instrukcji, które są w centrum zainteresowania takich firm, jak: Inmos, IBM, DEC, Hewlett-Packard. Konwencjonalne procesory są, zdaniem ich twórców, obarczone instrukcjami rzadko lub w ogóle nie używanymi. Kości można robić szybsze, tańsze i lepsze poprzez bardziej rozsądne ich projektowanie w myśl zasady, że szybciej wykona się pięć prostych instrukcji, niż trzy skomplikowane. Mikroprocesor typu RISC redukuje ilość instrukcji maszynowych do małej liczby najprostszych rozkazów, które dają się łatwo i szybko dekadować w procesorze.

Nie ulega wątpliwości, że z czasem ujrzą światło dzienne procesory o równoległym przetwarzaniu. Wyobraźmy sobie mikroprocesor złożony z wielu 32-bitowych kompletnych, niezależnych struktur porozumiewających się jakąś nadrzędną hierarchiczną siecią między sobą i wykonujących całkowicie równoległe operacje związane z realizacją danego programu. Takie procesory już się rodzą w laboratoriach. Jakie możliwości mógłby mieć taki procesor? Może nareszcie zdolny byłby zrozumieć ludzki język, a nie tylko w mniej lub bardziej udany sposób go naśladować.

Na horyzoncie pojawiają się też już „bioprocesory", choć narazie rozważania o nich przenieść należałoby ze sfery prognozowania w sferę „science-fiction". Chociaż gdy uświadomimy sobie, że narodziny pierwszego mikroprocesora odbyły się zaledwie przed 15 laty...

Krzysztof Czernek



TRZECI ODCINEK SŁOWNIKA DOTYCZY PRZETWARZANIA LIST, SŁÓW I LICZB.

3.1. Operacje arytmetyczne

ARCCOS n → n

Funkcja arcus cosinus. Jej wartością jest miara kąta w stopniach.

```
TO WIEL.CZEBYSZEWA :st :x
OUTPUT COS (:st   ARCCOS
:x )
END
PRINT WIEL.CZEBYSZEWA 5 0
0
PRINT WIEL.CZEBYSZEWA 5 -1
-1
```

ARCCOT n → n

Funkcja arcus cotangens. Jej wartością jest miara kąta w stopniach.

```
PRINT ARCCOT 0
90
PRINT ARCCOT 1
45
```

ARCSIN n → n

Funkcja arcus sinus. Jej wartością jest miara kąta w stopniach.

```
PRINT ARCSIN 0.5
30
PRINT ARCSIN 1
90
```

ARCTAN n → n

Funkcja arcus tangens. Jej wartością jest miara kąta w stopniach.

```
PRINT ARCTAN 0
0
PRINT ARCTAN 1 / SQRT 3
30
```

ATAN n → n

Patrz ARCTAN

COS n → n   COSINE n → n

Funkcja cosinus. Jej argumentem jest wartość kąta w stopniach.

```
PRINT COS 90
0
PRINT COS 60
0.5
```

COT n → n   COTANGENT n → n

Funkcja cotangens. Argumentem jest wartość kąta w stopniach.

```
PRINT COT 45
1
PRINT COT 0
COT doesn't like 0 as input
```

DIV n n → n

Iloraz dwóch liczb (pierwszego parametru przez drugi).

```
PRINT DIV 16 4
4
PRINT DIV 16 0
Can't divide by zero
```

INT n → n

Część całkowita liczby.

```
PRINT INT 6.789
6
PRINT INT -2.7
-3
```

INTEGER n → n

Patrz INT

PRODUCT n n → n   (PRODUCT n ...) → n

Iloczyn liczb, stanowiących parametry;

```
TO POTEGA :x :wykl
IF :wykl = 0 [OP 1]
OP PRODUCT :x POTEGA :x :
wykl - 1
END
PRINT POTEGA 2 16
65536
```

QUOTIENT n n → n

Patrz DIV

RANDOM n → n

Funkcja losowa o rozkładzie jednostajnym. Dla danego parametru naturalnego n wartością operacji jest losowa liczba naturalna z przedziału 0.. n-1 .

```
TO ZYGZAK :dlug
REPEAT :dlug [FD 10 +
RANDOM 90
LT RANDOM 360]
END
ZYGZAK 10
```

REMAINDER n n → n

Reszta z dzielenia pierwszego parametru przez drugi.

```
TO PARZYSTE :n
OP 0 = REMAINDER :n 2
END
PRINT PARZYSTE 666
TRUE
```

ROUND n → n

Zaokrąglenie parametru do najbliższej liczby całkowitej.

```
TO ZAOKR :x
OUTPUT (ROUND 100 * :x)
/ 100
END
PRINT ZAOKR 2 / 3
0.67
```

SIN n → n   SINE n → n

Funkcja sinus. Argumentem jest wartość kąta w stopniach.

```
PRINT SIN 180
0
PRINT SIN 30
0.5
```

SQRT n → n

Pierwiastek kwadratowy liczby nieujemnej, stanowiącej parametr.

```
PRINT SQRT 16
4
PRINT SQRT -3
SQRT doesn't like -3 as
input
```

SUM n n → n   (SUM n ...) → n

Suma liczb, stanowiących parametry;

```
TO SUMA :lista.skladn
IF EMPTYP :lista.skladn
[OP 0]
OP SUM LAST :lista.skladn
SUMA
BL :lista.skladn
END
PRINT SUMA READLIST
1 2 3 4 5 6 7 8
36
```

TAN n → n   TANGENT n → n

Funkcja tangens. Argumentem jest wartość kąta w stopniach.

```
PRINT TAN 45
1
PRINT TAN 90
TAN doesn't like 90 as
input
```

n * n → n
n / n → n
n + n → n
n − n → n

Cztery podstawowe operacje arytmetyczne: mnożenie, dzielenie, dodawanie i odejmowanie (wykonywane w tej kolejności). W celu zmiany kolejności wykonywania działań używa się nawiasów okrągłych.

```
PRINT 3 + 3 / 2
4.5
PRINT (3 + 3) / 2
3
```

3.2. Operacje logiczne

ALLOF p p → p   (ALLOF p ...) → p

Patrz AND

AND p p → p   (AND p ...) → p

Koniunkcja wyrażeń logicznych, stanowiących parametry operacji. (AND przyjmuje wartość TRUE, jeżeli wszystkie jego parametry mają wartość TRUE).

```
PRINT AND TRUE TRUE
TRUE
PRINT (AND TRUE FALSE
TRUE TRUE)
FALSE
```

ANYOF p p → p   (ANYOF p ...) → p

Patrz OR

NOT p → p

Zaprzeczenie wyrażenia logicznego, będącego parametrem.

```
TO MNIEJSZE.ROWNE :x :y
OUTPUT NOT :x > :y
END
PR MNIEJSZE.ROWNE 3 3
TRUE
```

OR p p → p   (OR p ...) → p

Alternatywa wyrażeń logicznych, stanowiących parametry operacji. (OR przyjmuje wartość FALSE, jeżeli wszystkie jego parametry mają wartość FALSE).

```
PRINT OR TRUE FALSE
TRUE
PR OR OR TRUE FALSE OR
FALSE FALSE
TRUE
```

Operatory służące do porównywania liczb (w przypadku znaku równości — dowolnych dwóch obiektów.

```
PRINT 2 = 3
FALSE
PRINT 1.6E12 < 1.5E13
TRUE
```

3.3. Zmienne

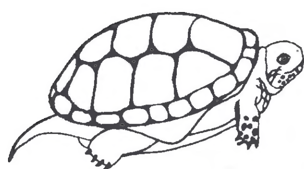
MAKE s nsl

Nadanie zmiennej o nazwie podanej jako pierwszy parametr wartości określonej przez parametr drugi. Jeżeli dana nazwa zmiennej pojawia się po raz pierwszy, tworzona jest nowa zmienna o tej nazwie, a następnie nadawana jej żądana wartość.

```
MAKE "x "iks
MAKE :x [120 130 140 150]
PRINT :x
iks
```



SŁOWNIK MINIMUM cz. III



```
PRINT :iks
120 130 140 150
```

NAMEP s → p

Operacja określająca, czy dane słowo jest nazwą istniejącej zmiennej.

```
MAKE "y 130
PRINT NAMEP "y
TRUE
PRINT NAMEP "namep
FALSE
```

THING s → nsl : s → nsl

Aktualna wartość zmiennej, której nazwa podana jest jako parametr.

```
MAKE "alfa "beta
MAKE :alfa "gamma
PRINT :alfa
beta
PRINT THING "alfa
beta
PRINT THING :alfa
gamma
PRINT THING THING "alfa
gamma
PRINT THING THING :alfa
gamma has no value
PRINT ::alfa
:alfa has no value
```

THING? s → p

Patrz NAMEP.

3.4. Listy i słowa

ASCII s → n

Numer podanego jako parametr znaku w kodzie ASCII.

```
TO SZYFR :slowo
IF EMPTYP :slowo [OP:slowo]
OP WORD CHAR 1 + ASCII
FIRST :slowo
SZYFR BF :slowo
END
PR SZYFR "mikrokomputer
njlsplpnqvufs
```

BUTFIRST sl → sl BF sl → sl

Operacja, której wynikiem jest podany jako parametr obiekt bez pierwszego elementu.

```
PRINT BUTFIRST "ucho
cho
PRINT BF [Dzien dobry !]
dobry !
PRINT BUTFIRST []
BUTFIRST doesn't like []
as input
```

BUTLAST sl → sl BL sl → sl

Operacja, której wynikiem jest dany obiekt bez ostatniego elementu.

```
TO ROZWIN :obiekt
IF EMPTYP :obiekt [STOP]
ROZWIN BUTLAST :obiekt
PRINT :obiekt
END
ROZWIN "kwiat
k
kw
kwi
kwia
kwiat
```

CHAR n → s

Znak kodu ASCII odpowiadający podanej jako parametr liczbie.

```
TO DUZE :slowo.
IF EMPTYP:slowo [OP:slowo]
MAKE "znak FIRST :slowo
MAKE "numer ASCII :znak
OP WORD(IF AND(96<:numer)
(123>:numer)[CHAR:numer-32]
[:znak]) DUZE BF :slowo
END
PRINT DUZE "Malutkie
MALUTKIE
```

COUNT sl → n

Ilość elementów danego obiektu.

```
PRINT COUNT [1 2 3 4 5]
5
PRINT COUNT 1234
4
PRINT COUNT SE 1 "
2
PRINT COUNT []
0
```

EMPTYP sl → p

Operacja określająca, czy dany obiekt jest pusty (zawiera 0 elementów).

```
PRINT EMPTYP 0
FALSE
PRINT EMPTYP SE [] []
TRUE
PRINT EMPTYP SE [] "
FALSE
PRINT EMPTYP "
TRUE
```

EQUALP nsl nsl → p

Operacja określająca, czy dane dwa obiekty są równe.

```
PRINT EQUALP TRUE "TRUE
TRUE
PRINT EQUALP TRUE "true
FALSE
PR EQUALP [1 []] SE 1 []
FALSE
```

FIRST sl → sl

Operacja, której wynikiem jest pierwszy element podanego jako parametr obiektu.

```
PRINT FIRST "LOGO
L
PRINT FIRST [Q W E R T Y]
Q
PRINT FIRST []
FIRST doesn't like [] as input
```

FPUT nsl l → l

Operacja, której wynikiem jest lista utworzona poprzez włączenie obiektu podanego jako parametr pierwszy na początek listy określonej przez parametr drugi.

```
TO PRZEWINIETA :lista
OP FPUT LAST :lista BL
:lista
END
PRINT PRZEWINIETA [alfa
beta gamma]
gamma alfa beta
```

ITEM n l → nsl

Operacja, której wynikiem jest określony przez pierwszy parametr element listy podanej jako parametr drugi.

```
TO ELT.LOSOWY :lista
OP ITEM 1 + RANDOM COUNT
:lista
:lista
END
REPEAT 3 [PRINT ELT.
LOSOWY [a b
c d]]
a
d
b
```

LAST sl → sl

Operacja, której wynikiem jest ostatni element podanego jako parametr obiektu.

```
TO ZENSKI :rzeczownik
OP MEMBERP LAST
:rzeczownik [A a]
END
PRINT ZENSKI "szafa
TRUE
PR ZENSKI "krzeslo
FALSE
```

LIST nsl nsl → l (LIST nsl ...) → l

Operacja tworząca listę, zawierającą podane jako parametry obiekty.

```
SHOW LIST 1 2
[1 2]
SHOW (LIST [] 2 [])
[[] 2 []]
```

LISTP nsl → p

Operacja określająca czy dany obiekt jest listą.

```
PRINT LISTP "Logo
FALSE
PRINT LISTP [Logo]
TRUE
```

LIST? nsl → p

Patrz LISTP

LPUT nsl l → l

Operacja, której wynikiem jest lista powstała przez dołączenie obiektu określonego przez parametr pierwszy na koniec listy podanej jako parametr drugi.

```
PRINT LPUT "dobry [Dzien]
Dzien dobry
PRINT LPUT [a b] [c d e]
c d e [a b]
```

MAMBERP nsl l → p

Operacja określająca, czy obiekt podany jako parametr pierwszy jest elementem listy określonej przez parametr drugi.

```
PRINT MEMBERP [] []
FALSE
PR MEMBERP " []
FALSE
PRINT MEMBERP [] [[]]
TRUE
PRINT MEMBERP 12 [12 13 14]
TRUE
```

NUMBERP nsl → p

Operacja określająca, czy dany obiekt jest liczbą.

```
PRINT NUMBERP 2.78E9
TRUE
PRINT NUMBERP "232
TRUE
PRINT NUMBERP WORD 2 ".05
TRUE
```

NUMBER? nsl → p

Patrz NUMBERP

SENTENCE nsl nsl → l (SE nsl ...) → l

Operacja tworząca listę z podanych jako parametry obiektów. Jeżeli któryś z parametrów jest listą, jej zewnętrzne nawiasy są usuwane.

```
SHOW SENTENCE 1 2
[1 2]
SHOW (SE [] 2 [])
[2]
```

WORD s s → s (WORD s ...) → s

Operacja tworząca słowo złożone z podanych parametrów.

```
TO SKLEJONE :lista
IF EMPTY :lista [OP " ]
OP WORD FIRST :lista
SKLEJONE BF
:lista
END
PR SKLEJONE [mikro komputer
-1 986]
mikrokomputer-1986
```

WORDP nsl → p

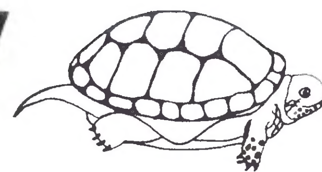
Operacja określająca, czy dany obiekt jest słowem.

```
PRINT WORDP "alfa
TRUE
PRINT WORD 234
TRUE
PRINT WORDP TRUE
TRUE
PRINT WORDP [alfa beta]
FALSE
```

WORD? nsl → p

Patrz WORDP

Marcin Waligórski



```
to suma :a :b
if :b = zero [op :a]
op dodane1 suma :a odjete1 :b
end

to iloczyn :a :b
if :b = zero [op zero]
op suma :a iloczyn :a odjete1 :b
end

to potega :a :n
if :n = zero [op jeden]
op iloczyn (potega :a odjete1 :n) :a
end

to zero
op 0
end

to jeden
op 1
end

to dodane1 :a
op :a + 1
end

to odjete1 :a
op :a - 1
end

?pr suma 10 7
17
?pr suma 52 15
67
?pr iloczyn 8 6
48
?pr iloczyn 18 0
0
?pr potega 10 0
1
?pr potega 2 4
16
```

```
to zero
op "
end

to jeden
op "#
end

to dodane1 :a
op word "# :a
end

to odjete1 ;a
op bf :a
end

?
?pr suma "### "#####
########
?pr suma "###### "#
#######
?pr iloczyn "##### "###
###############
?pr iloczyn "######## "#
########
?pr potega "########## "
#
?pr potega "### "###
###########################
```

LICZBY NATURALNE INACZEJ

Co to są liczby naturalne — wiemy wszyscy. Liczby te mogą być wprowadzone aksjomatycznie, tzn. można przyjąć jako aksjomaty kilka własności tych liczb. Pozostałe ich własności można wyprowadzić z przyjętych aksjomatów.

Przyjmujemy zatem następujące aksjomaty:

1. Każda liczba naturalna posiada dokładnie jeden następnik (liczbę następną względem danej).
2. Istnieje liczba naturalna — nazywamy ją zerem — która nie jest następnikiem żadnej liczby naturalnej.
3. Różnym liczbom naturalnym odpowiadają różne następniki.
4. Aksjomat ostatni to dobrze znana zasada indukcji matematycznej, której w pełnym brzmieniu przytaczać nie będziemy (można zajrzeć do podręcznika matematyki).

Przy pomocy powyższych aksjomatów (pochodzących od Peano), możemy łatwo zdefiniować podstawowe działania arytmetyczne. Oznaczymy w tym celu następnik liczby a jako a + 1, zaś poprzednik tej liczby (tj. liczbę, której następnikiem jest a) jako a − 1.

Zdefiniujmy dodawanie jako

$$a + b = \begin{cases} a & \text{dla } b = 0 \\ (a + (b-1)) + 1 & \text{dla } b \neq 0 \end{cases}$$

Dodawanie sprowadza się w naszym pojęciu do wielokrotnych operacji brania następnika liczby a (lepiej to dokładnie przeanalizować!). Z kolei mnożenie możemy zdefiniować, jako wielokrotne dodawanie

$$a \times b = \begin{cases} 0 & \text{dla } b = 0 \\ a + a \times (b-1) & \text{dla } b \neq 0 \end{cases}$$

Wprowadzając pojęcie liczby 1, zdefiniowanej jako 1 = 0 + 1 możemy zdefiniować również potęgowanie jako wielokrotne mnożenie.

Napisane w LOGO procedury **suma, iloczyn, potęga** stanowią realizację wprowadzonych przez nas definicji.

Zauważmy teraz, że same liczby naturalne możemy reprezentować w dość dowolny sposób — należy jedynie określić wyróżnione przez nas liczby 0 i 1 oraz zdefiniować, zgodnie z przyjętymi aksjomatami, poprzednik i następnik liczby.

Pierwszy przykład zdefiniowania procedur **zero, jeden, dodane 1** i **odjęte 1** pozwala przekonać się, że działania arytmetyczne określiliśmy poprawnie.

W przykładzie drugim reprezentujemy liczbę jako ciąg krzyżyków o odpowiedniej długości. Przykłady wykonanych obliczeń przekonują nas, że jest to reprezentacja liczb równoważna poprzedniej.

Wreszcie bardziej złożony przykład: zapisuje liczby w postaci rzymskiej. Nowe procedury **zero, jeden, dodane 1, odjęte 1** i dodatkowo **uproszczone, skomplikowane, kon i nowykoniec** pozwalają dokonywać obliczeń na liczbach rzymskich. Ciekawym szczegółem tej metody, wynikającym wprost z przyjętych założeń, jest to, że w żadnym miejscu nie uciekamy się do obliczeń innych, niż operacje na słowach.

Warto samemu popróbować dalej. Dobrym ćwiczeniem może być podobna realizacja działań arytmetycznych na liczbach w postaci dwójkowej. Powodzenia.

na podstawie pracy: H.D. Boecker „Functional Programming in Basic-Plus"

opracował Marcin Waligórski

```
to zero
op "
end

to jeden
op "I
end

to dodane1 :a
op uproszczone word :a "I
end

to odjete1 :a
op bl skomplikowane :a
end

to kon :slowo :koncowka
if emptyp :koncowka [op "TRUE]
if emptyp :slowo [op "FALSE]
op and (equalp last :slowo last :koncowk
a) (kon bl :slowo bl :koncowka)
end

to nowykoniec :dl :slowo :koniec
make "x :slowo
repeat :dl [make "x bl :x]
op word :x :koniec
end

to uproszczone :a
if kon :a "IIII [op uproszczone nowykoni
ec 4 :a "IV]
if kon :a "IVI [op uproszczone nowykonie
c 3 :a "V]
if kon :a "VIV [op uproszczone nowykonie
c 3 :a "IX]
if kon :a "IXI [op uproszczone nowykonie
c 3 :a "X]
if kon :a "XXXX [op uproszczone nowykoni
ec 4 :a "XL]
if kon :a "XLIX [op uproszczone nowykoni
ec 4 :a "IL]
if kon :a "ILI [op uproszczone nowykonie
c 3 :a "L]
if kon :a "LXL [op uproszczone nowykonie
c 3 :a "XC]
if kon :a "XCIX [op uproszczone nowykoni
ec 4 :a "IC]
if kon :a "ICI [op uproszczone nowykonie
c 3 :a "C]
op :a
end

to skomplikowane :a
if kon :a "IV [op nowykoniec 2 :a "IIII]

if kon :a "V [op nowykoniec 1 :a "IVI]
if kon :a "IX [op skomplikowane nowykoni
ec 2 :a "VIV]
if kon :a "X [op nowykoniec 1 :a "IXI]
if kon :a "XL [op skomplikowane nowykoni
ec 2 :a "XXXX]
if kon :a "IL [op skomplikowane nowykoni
ec 2 :a "XLIX]
if kon :a "L [op nowykoniec 1 :a "ILI]
if kon :a "XC [op skomplkwane nowykoniec
 2 :a "LXL]
if kon :a "IC [op skomplikowane nowykoni
ec 2 :a "XCIX]
if kon :a "C [op nowykoniec 1 :a "ICI]
op :a
end

?
?pr suma "IV "II
VI
?pr suma "IX "VI
XV
?pr iloczyn "X "

?pr iloczyn "IV "II
VIII
?pr potega "V "
I
?pr potega "IV "II
XVI
```

ZX SPE

W każdym mikrokomputerze możemy wyodrębnić kilka bloków funkcjonalnych, które połączone w odpowiedni sposób tworzą tzw. system czyli to, z czym ma do czynienia użytkownik. Zwykle podstawowe bloki funkcjonalne zamknięte są we wspólnej obudowie wraz z klawiaturą. Jeśli zajrzymy do środka, zobaczymy kilkanaście lub kilkadziesiąt układów scalonych różnej wielkości, zmontowanych na wspólnej płytce o skomplikowanym obwodzie drukowanym, umieszczonym na obu jej stronach, oraz niewielką ilość elementów pomocniczych. Większe mikrokomputery posiadają kilka płytek, małe — tylko jedną. Płytka naszego Spectrum zawiera 25 układów scalonych.

Elektronik obeznany z techniką mikrokomputerową, z łatwością wyodrębni poszczególne bloki tworzące system mikrokomputerowy. Zgodnie z zasadami obowiązującymi w informatyce będą to:

1. mikroprocesor — zwany też z angielska CPU (Central Processing Unit);
2. pamięć operacyjna podzielona na pamięć ROM (Read Only Memory) — tylko do odczytu i RAM (Random Access Memory) — pamięć o swobodnym dostępie;
3. urządzenia wejścia/wyjścia zwane portami lub bramami. W naszym Spectrum wszystkie te funkcje łączy w sobie jeden układ scalony zwany ULA (Uncommited Logic Array), co po polsku można wyłożyć, jako rozsyłająca, rozdzielająca matryca logiczna. Wszystkie te układy zrealizowane są w technologii o wysokiej skali integracji (LSI).

Oprócz nich na płytce znajdują się układy zrealizowane w technologii o małej lub średniej skali integracji tzw. układy logiczne TTL. Realizują one funkcje pomocnicze, głównie jako układy sterujące blokiem pamięci RAM.

W skomplikowanej mozaice połączeń między wyprowadzeniami układów scalonych informatyk rozpozna tzw. magistrale. Są to: magistrala adresowa i magistrala danych oraz magistrala sterująca. Wszystkie one oraz kilka dodatkowych, charakterystycznych dla Spectrum, wyprowadzone są na złącze krawędziowe znajdujące się z tyłu mikrokomputera i pozwalające na współpracę z dodatkowymi urządzeniami peryferyjnymi, jak np. drukarka, Interface 1 itp. Na płytce są również umieszczone złącza służące do przyłączenia odbiornika TV jako monitora, magnetofonu jako pamięci masowej oraz klawiatury. Są to podstawowe urządzenia peryferyjne mikrokomputera ZX SPECTRUM tworzące wraz z nim kompletny system mikrokomputerowy.

W ZX SPECTRUM zastosowano mikroprocesor opracowany w firmie Zilog — Z80 A. Jest to 8-bitowy mikroprocesor pracujący z zegarem 3,5 MHz z 16-bitową magistralą adresową. Umożliwia on zaadresowanie 64 kB pojedynczych komórek pamięci operacyjnej. Dokładnie może być tych komórek 65 536. Do każdej z nich można zapisać lub odczytać z niej 256 różnych wartości od 0 do 255. Oczywiście oprócz pamięci procesor musi obsługiwać inne bloki zwane urządzeniami wejścia-wyjścia. Aby jednoznacznie określić, które bloki mają być aktywne, procesor zgodnie z programem ustawia odpowiednią kombinację sygnałów na magistrali sterującej. Jasne jest, że urządzenia wejścia-wyjścia mają również swoje adresy po to, aby np. sygnał przeznaczony dla monitora nie trafił przez przypadek do magnetofonu lub odwrotnie.

Spójrzmy teraz na rys. 2. W Spectrum obszar od adresu 0 do adresu 16 383 zarezerwowany jest dla pamięci ROM. W ROM-ie przechowywany jest interpretator języka BASIC, program zwany systemem operacyjnym oraz program obsługujący monitor. Poczynając od adresu 16 384 aż do 65 535 rozciąga się pamięć RAM. W Spectrum 16 k ostatnim adresem w pamięci jest 32 767.

Pierwsze 6 192 bajty RAM zarezerwowane są dla pamięci ekranu. Dalsze 2 000 bajtów zarezerwowane są dla systemu. W obszarze tym komputer przechowuje niezbędne do pracy informacje o stanie systemu.

Poczynając od adresu ok. 25 000 rozciąga się obszar pamięci dostępny dla użytkownika. Przechowywany jest w niej program w BASIC-u lub w kodzie maszynowym. Pod samym szczytem pamięci (ang. RAMTOP) istnieje obszar, w którym można przechowywać znaki graficzne, możliwe do zdefiniowania przez użytkownika za pomocą odpowiednich instrukcji (tzw. UDG).

Aby móc komunikować się z komputerem lub choćby uruchomić jakąkolwiek grę, potrzebny jest monitor i magnetofon oraz klawiatura czyli urządzenia peryferyjne. Obsługą tych urządzeń zajmuje się wspomniany wyżej ULA, który spełnia następujące funkcje: tworzy obraz telewizyjny łącznie z sygnałami koloru, przyjmuje i wysyła sygnał z programem przeznaczonym dla magnetofonu, generuje jednogłosowy dźwięk w zakresie jedenastu oktaw, wytwarza sygnał zegarowy dla mikroprocesora i przede wszystkim obsługuje i odświeża pamięć obrazu, a także 50 razy w ciągu sekundy kontroluje stan klawiatury. Widać stąd, że bez układu ULA nasz Spectrum byłby ślepy i głuchy, a my niewiele moglibyśmy się dowiedzieć o tym, co się dzieje w jego wnętrzu.

Wszystkie te bloki aby działać muszą być zasilane. Do tego celu służy kilkadziesiąt małych elementów, zmonto-



FOT. L. DZIKOWSKI

ZX SPECTRUM 48 KB



wanych na wspólnej płytce komputera. Zamieniają one napięcie doprowadzone z oddzielnego zasilacza na stabilizowane napięcie potrzebne do prawidłowej pracy wszystkich układów scalonych. Niestety ten wewnętrzny zasilacz Spectrum jest jego najsłabszą stroną.

Oprócz zasilacza na płytce znajdują się jeszcze: koder PAL — tworzący barwny sygnał telewizyjny w standardzie PAL oraz modulator UHF, pozwalający uzyskać obraz na ekranie telewizora dostrojonego w pobliże 36 kanału.

Parametry użytkowe ZX Spectrum nie są, jak na 1986 rok, olśniewające. Trzeba pamiętać jednak, że jest to komputer, który pojawił się na rynku 4 lata temu, kiedy nikt nie słyszał jeszcze o firmie Amstrad, a Commodore czy Atari dopiero miały nadzieję na udany model swojego mikrokomputera.

Z punktu widzenia programisty, podstawową wadą Spectrum jest dość skomplikowany sposób korzystania z pamięci operacyjnej. Spowodowane jest to rozdzieleniem pamięci RAM na dwa bloki 16 i 32 k. Blok 32 k jest przyłączony bezpośrednio do mikroprocesora, natomiast dostęp do bloku 16 K zawierającego pamięć obrazu, zmienne systemowe itd. sterowany jest przez układ ULA. Wynikają z tego dość istotne ograniczenia w działaniu systemu. Mianowicie zapis do pamięci 16 k może odbywać się tylko wtedy, kiedy nie korzysta z niej ULA. Wprawdzie dzieje się to około 50 razy na sekundę, ale przez bardzo krótki okres, podczas którego procesor musi wpisać do pamięci nowe dane, odczytać stan zmiennych systemowych, stosu itd. Ma na to wszystko około 2 ms. Powoduje to bardzo powolne działanie programu, który wymaga dostępu do pamięci ekranu.

Poniższy program stanowi prosty przykład ilustrujący szybkość działania:

```
1ϕ FOR u = 32768 TO 33768
15 PRINT AT 0,0;n
2ϕ POKE n,255
3ϕ NEXT n
```

Czas realizacji wynosi ok. 24 sekund. Po wyrzuceniu linii 15 program wykonuje się w 6 sekund czyli 4 razy szybciej. Różnica jest tym większa, im większy jest obszar ekranu, z którego korzystamy. Drugą wadą naszego Spectrum jest fakt, iż procesor zostaje zatrzymany podczas generacji dźwięku, co również powoduje wydłużenie działania programu.

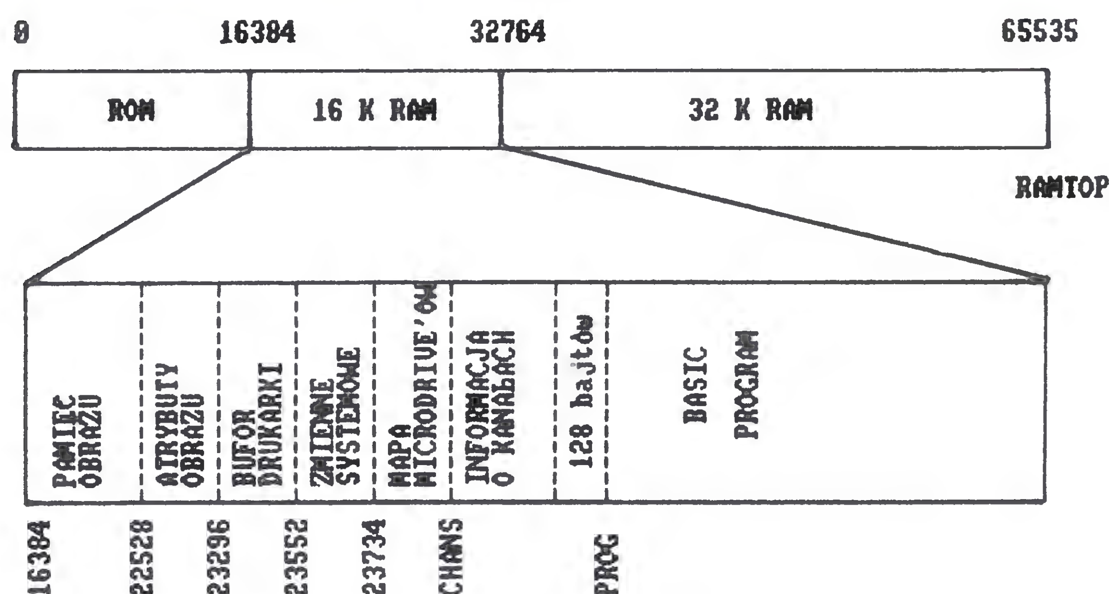
Poza opisanymi wadami, a raczej niedogodnościami strukturalnymi, należy wspomnieć o błędach zawartych w programie operacyjnym. Wynikły one głównie z pośpiechu przy projektowaniu systemu. Najważniejszy z nich to błąd w procedurze obsługi przerwań. Ponieważ błędy te są istotne tylko dla zaawansowanych programistów, korzystających z procedur zawartych w ROM-ie, nie będziemy ich szerzej omawiać.

Na koniec jeszcze kilka słów o wadach technologicznych czyli wynikających z rodzaju zastosowanych materiałów oraz rozwiązań konstrukcyjnych. I tu na czoło wysuwa się klawiatura. Jest ona wykonana z dwuwarstwowej folii z napylonymi próżniowo polami kontaktowymi i ścieżkami połączeń. Klawiatura stanowi „piętę achillesową" Spectrum. Niestety, firma Sinclair nie potrafiła rozwiązać tego problemu. Klawiatury modelu Spectrum + poza tym, że są wygodniejsze, nadal pozostawiają wiele do życzenia, jeśli idzie o trwałość. Podstawowymi uszkodzeniami są przerwy i zwarcia, które uniemożliwiają pracę komputera.

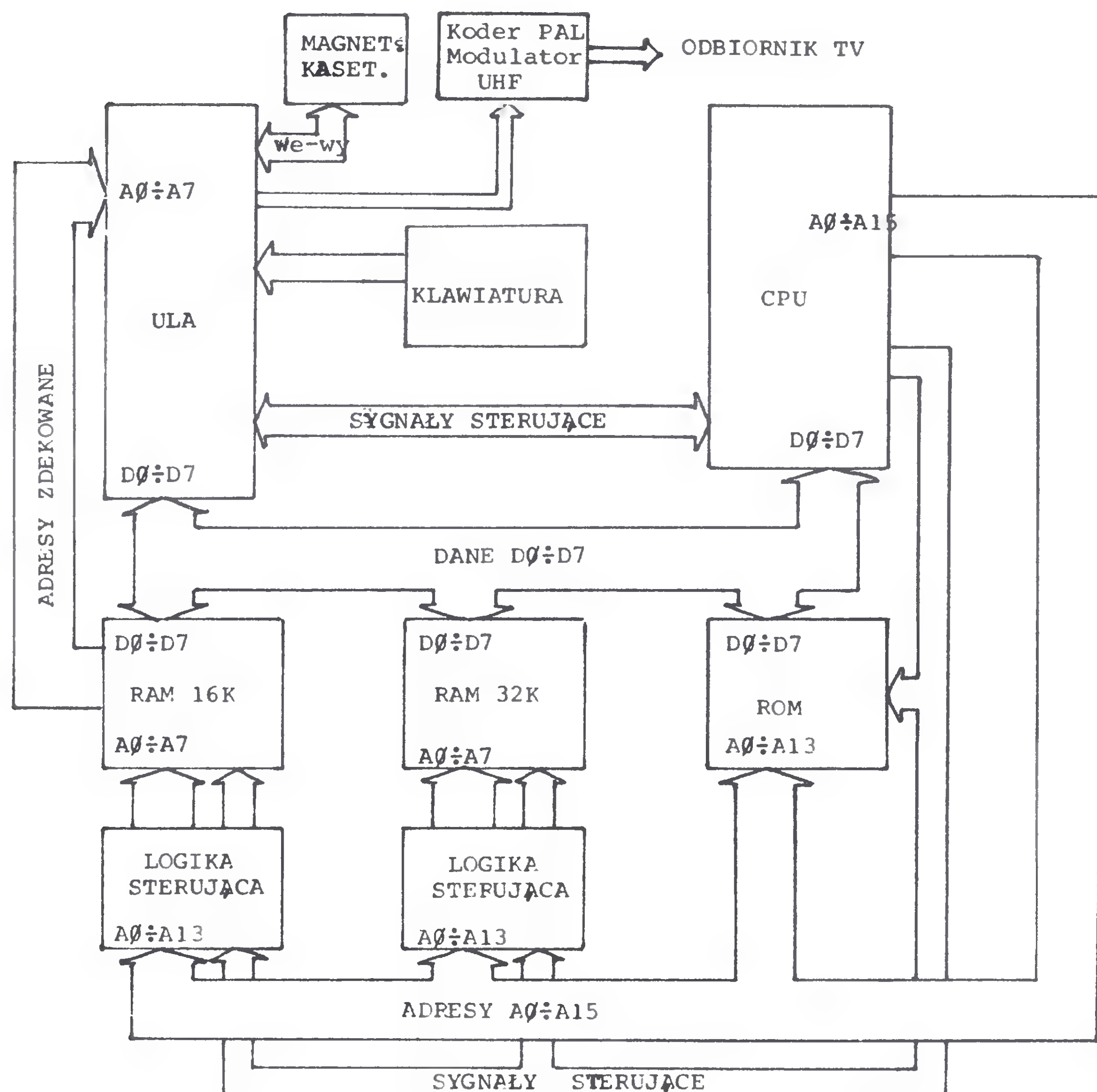
Drugą notoryczną wadą są uszkodzenia zasilacza, szczególnie przetwornicy zasilającej pamięci 16 K.

Mimo niewątpliwych wad, ma też Spectrum swoje zalety. Przede wszystkim jest to komputer nadal najbardziej dostępny w naszym kraju. Istnieje bogata literatura opisująca zarówno sprzęt jak i oprogramowanie. Ceny urządzeń peryferyjnych nie osiągają astronomicznych sum, a co najważniejsze jest tych urządzeń sporo.

Grzegorz Grzybowski



RYS. 2: Mapa pamięci



RYS.1: SCHEMAT BLOKOWY ZX SPECTRUM 48K

WŁASNE LITERY

Zapewne każdy użytkownik komputera ZX SPECTRUM po przeczytaniu tego tytułu pomyśli z niechęcią: „Tak, tak, wiem! SPECTRUM ma możliwość definiowania 21 własnych znaków graficznych. Nic ciekawego!". Otóż nie, chcemy zaproponować wam coś innego.

SPECTRUM ma możliwość zdefiniowania całego alfabetu. Możesz zastosować własny krój pisma, cyfr, znaków specjalnych! Więcej — możesz zmienić krój pisma w trakcie wykonywania programu — i to w każdej chwili.

Jak Ci wiadomo każdy znak to 64 punkty w kolorach atramentu lub tła. 64 punkty to 64 bity, czyli 8 bajtów. Wystarczy zatem przepisać odpowiednio 8 bajtów w pamięć obrazu komputera, aby na ekranie ukazała się mozaika tworząca znak, taka, jak na rysunku. Tam, gdzie były jedynki stawiany jest punkt w kolorze atramentu, tam gdzie zera — w kolorze tła. W pamięci **ROM** znajdują się gotowe „mozaiki", czyli bajty odpowiadające znakom. Obejmują one wszystkie znaki począwszy od spacji (odstęp) aż do znaku „copyright" (literka „c" w kółeczku). Umieszczone są one po kolei zaczynając od adresu 15616 i zajmują 768 bajtów. Adres początku tego obszaru (pomniejszony o 256) zapisany jest w pamięci RAM, w jednej ze zmiennych systemowych o nazwie **CHAR$** (adres 236ø6 i 236ø7). Zapytasz zapewne, czy nie można go było umieścić na stałe w pamięci **ROM**. Oczywiście, można było, ale umieszczono go w RAM po to, abyś mógł go zmienić i w ten sposób definiować własne znaki wyświetlane przez komputer.

Aby zdefiniować „własny generator znaków" należy zrobić miejsce na nowy wzorzec (obniżając **RAMTOP**), wpisać wzorzec w to miejsce, a na koniec zmienić wartość zmiennej CHAR$, aby komputer pobierał bajty wzorca definiowane przez nas, czyli z naszego obszaru.

```
10 LET RAMTOP = PEEK 23730 + 256 * PEEK
23731
20 CLEAR RAMTOP - 768
30 LET NOWY = PEEK 23730 + 256 * PEEK 22
3731 + 1
40 FOR I = 0 TO 767 : POKE NOWY + I , PE
EK ( 15616 + I ) : NEXT I
50 LET NOWY = NOWY - 256 : PRINT NOWY
60 POKE 23606 , NOWY -256 * INT (NOWY /
256 ) : POKE 23607 , INT ( NOWY / 256 )

READY.
```

W linii 1ø komputer na podstawie zmiennej systemowej **RAMTOP** (umieszczonej w komórkach 2373ø i 23731) określa, czy jego pamięć **RAM** ma 16 czy 48 kB, a następnie robi miejsce na nową tablicę wzorców (instrukcja **CLEAR**) obniżając **RAMTOP**. Linia 3ø ustala adres początku naszego obszaru wzorców (wpisując go pod zmienną „nowy"), a linia 4ø przepisuje kolejno zawartość „fabrycznego" obszaru wzorców do „naszego". Linia 5ø określa wartość, którą trzeba wpisać w komórki **CHAR$**, a linia 6ø realizuje to.

Wpisz program do komputera i uruchom go. Początkowo komputer „zamilknie" na kilka sekund (w tym czasie przepisuje „fabryczny" wzorzec do „naszego"), następnie wyświetli cyfrę (zapisz ją!) i zgłosi się normalnym raportem u dołu ekranu. Jednakże od tej chwili wyświetla on już nie znaki „fabryczne", ale ukryte w pamięci **RAM** od adresu, który wyświetlił. Dzięki linii 4ø są one w tej chwili takie same, jak „fabryczne", ale nic nie stoi na przeszkodzie, aby je modyfikować. Możesz użyć do tego celu programu:

```
10 INPUT " ZNAK " ;A$
20 LET ACODE = CODE A$
30 LET ADRES = PEEK 23606 + 256 * PEEK 2
3607 + 256 + 8 * ( ACODE - 32 )
40 FOR I = ADRES TO ADRES + 7 : INPUT "
LINIA " ; ( I - ADRES + 1 ) ; " ? " ; LI
NIA
50 POKE I , LINIA : PRINT AT 10 , 16 ; A
$ : NEXT I
```

Do wpisywania kolejnych bajtów wygodnie jest posługiwać się instrukcją **BIN**, gdyż operuje ona na poszczególnych bitach i „widać" wtedy, co wpisujemy.

Komputer z „nową" tablicą wzorców zachowuje się dokładnie tak, jak z „fabryczną" (pomijając treść wyświetlanych znaków) do momentu zmiany wartości zmiennej systemowej **CHAR$** (można to zrobić nieświadomie używając instrukcji **NEW**). Na szczęście nie wszystko jest wówczas stracone — „nasz" wzorzec został w pamięci i aby go uaktywnić wystarczy wpisać adres „naszego" wzorca zmniejszony o 256 do komórek 236ø6 i 236ø7. Wartość ta jest wyświetlana przez komputer w trakcie realizacji programu (dlatego radziłem zapisać!), a zapis realizuje się tak, jak w linii 6ø; należy tylko pamiętać, aby obie wartości wpisać do pamięci jednocześnie. Jeżeli wpiszemy tylko jeden bajt zmiennej, komputer potraktuje to jak zdefiniowanie nowego adresu obszaru wzorców i wyświetlane znaki przybiorą całkowicie przypadkowe kształty.

Pracowicie zdefiniowany alfabet można nagrać na taśmę za pomocą instrukcji **SAVE** (nazwa) **CODE** liczba, 768, gdzie nazwę wybieramy dowolnie, a liczba to początek tablicy wzorca, który komputer nam wyświetlił (a który mamy zapisany na kartce). Chcąc wykorzystać „nasz" alfabet w programie, na samym jego początku piszemy linię

1 CLEAR liczba — 1 LOAD " "CODE : POKE 236ø6, liczba — 256 * INT (liczba) (256) : POKE 236ø7, INT (liczba) (256)

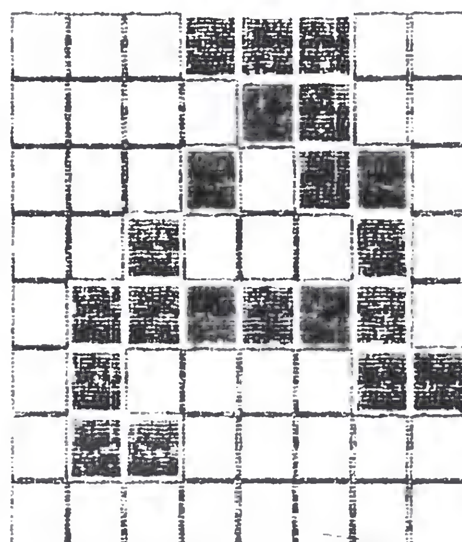
za słowo „liczba" wstawiając początek wzorca. Wzorzec nagrywamy bezpośrednio za programem, który sam go sobie wczyta i przyłączy.

Na zakończenie przyjrzyj się przykładowemu programowaniu gotyckiej litery „A" (patrz rys. 1), którą umieścimy w miejscu normalnej litery „A" (wyrównać wszystkie **ENTER**

znak? A	ENTER
linia1? BIN 111øø	ENTER
linia2? BIN 11øø	ENTER
linia3? BIN 1ø11ø	ENTER
linia4? BIN 1øøø1ø	ENTER
linia5? BIN 111111ø	ENTER
linia6? BIN 1øøøø11	ENTER
linia7? BIN 11øøøøø	ENTER
linia8? ø	ENTER

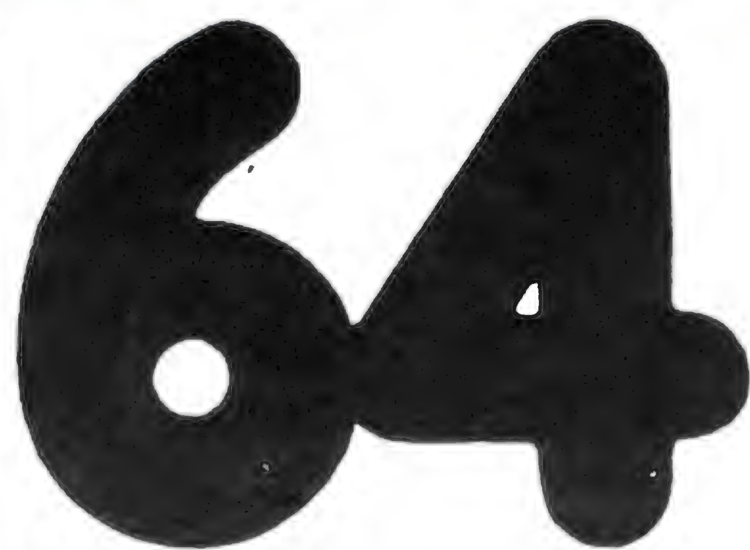
i od tej chwili znak „A" wyświetlany będzie innym krojem pisma.

Dariusz A. Przygoda



= 00011100
= 00001100
= 00010110
= 00100010
= 01111110
= 01000011
= 01100000
= 00000000

Obraz gotyckiej litery „A"



KOLUMNY TEKSTU

Program ten (1,3 kB),przeznaczony dla **ZX Spectrum 48 kB** umożliwia uzyskanie na ekranie 64 kolumn tekstu zamiast standardowych 32. Pozwala na to dodatkowy rozkaz **TAB** (EXTENDED MODE, P) o formacie prawie takim samym jak rozkaz **PRINT.** Umożliwia on drukowanie zarówno tekstów, jak i liczb w dowolnych kombinacjach używając operacji **AT,** średników, przecinków, funkcji, rozkazów zmieniających kolor itp. Np.:

1øTAB "Mamy 64 kolumny tekstu"
2øTAB INK 4;PAPER 7;PI:LET A$="ABCD"
3øTAB AT 1ø,5ø;SIN 5.2,98765;A$

Równolegle z rozkazem TAB można używać standardowego PRINT.

Np.

1øTAB "Tak drukuje nowy rozkaz"
2ø PRINT "a tak PRINT"

Należy zwrócić uwagę, że nasz rozkaz TAB jest czym innym niż operacja TAB używana w BASIC-Spectrum do określania kolumny druku z rozkazem PRINT. Mamy więc dwa różne TAB, jednakże standardową operację TAB można używać także z naszym rozkazem TAB, np.

1øTAB TAB 52;"QWERTY"

Rozkaz TAB jest szczególnie wygodny, gdy np. chcemy wypisać dane liczbowe w pięciu kolumnach, lub gdy rysujemy wykresy i chcemy je później opisać. Litery i cyfry używane przez Spectrum są zazwyczaj za duże do tego celu. Są dwie zasadnicze różnice w działaniu TAB i PRINT. Pierwszy nie drukuje znaków o CHR$ ø do 31 (co właściwie nie jest istotne), rozkazów BASIC-u oraz **UDG.** Wszystkie symbole **UDG** pozostają dostępne użytkownikowi wg jego życzenia, lecz poprzez PRINT. Zamiast znaków wpisywanych modem graficznym (CAPS SHIFT 9) są zdefiniowane polskie litery ą,ć,ę,ł,ń,ś,ó,ż oraz kilka symboli matematycznych — patrz tabela.

Oczywiście symbole te pojawiają się na ekranie dopiero po wykonaniu danego rozkazu. W listingu zamiast ą,ę, itd. będą widoczne normalne symbole graficzne.

Drugą cechą różniącą TAB od PRINT jest niezupełne działanie OVER. Aby otrzymać poprawne wyniki używając tego rozkazu razem z naszym TAB należy pamiętać o pisaniu w co drugiej kolumnie.

Uruchomienie programu

1. Po włączeniu komputera wpisać program DATA.
2. Nagrać go na taśmę rozkazem **SAVE"DATA"**
3. Wykonać rozkaz NEW i wpisać: **1ø CLEAR 64øø1:LOAD""CODE:RANDOMIZE USR 64øø2:LOAD""**
4. Nagrać na taśmę rozkazem **SAVE" 64 kolumny"LINE1ø**
5. Odegrać z taśmy program DATA rozkazem **LOAD"",** sprawdzić czy nie zawiera błędów i uruchomić go rozkazem RUN
6. Po raporcie **OK** nagrać zapisany w ten sposób kod maszynowy rozkazem **SAVE"CODE"CODE" 64øø2,1366** i sprawdzić poprawność rejestracji na taśmie poprzez **VERIFY""CODE**
7. Wykonać **NEW** po czym wpisać **1 POKE 63998,26:POKE 63999,25ø** i zastartować poprzez **RUN**
8. Jeżeli nie zostanie wyświetlony raport **OK** lub po wpisaniu dowolnego z przykładowych rozkazów program nie będzie działał — wyłączyć komputer i powrócić do punktu 5, — DATA zawiera błędy niezauważone przy wpisywaniu.
9. Wpisać **POKE 23756,ø** w ten sposób linia 1 zostanie zabezpieczona przed przypadkowym skasowaniem i nagrać na taśmę linię 1 rozkazem **SAVE"64"LINEø**
10. Pierwszy z nagranych czterech programów tzn. DATA jest już niepotrzebny, można go ewentualnie skasować. Pozostałe trzy nagrania stanowią wersję ostateczną programu. Można go nagrywać poprzez **LOAD"64 kolumny"** lub po prostu LOAD""

I to wszystko. Zwracamy uwagę, że aby program działał poprawnie przy każdym użyciu **RUN** musi on przejść przez linię ø przed dojściem do jakiegokolwiek rozkazu TAB (tzn. np. **RUN 5** jest niedopuszczalne, ale można, po przynajmniej jednokrotnym przejściu przez linię ø, użyć **GOTO 5/.** W przeciwnym wypadku wyświetlony zostanie raport: Nonsence in BASIC.

Życzymy przyjemnej zabawy i na początek proponujemy wpisanie:

1ø FOR n=32 TO 143:TAB CHR$ n;:NEXT n

Andrzej Zagrodziński

Numer	1	2	3	4	5	6	7	8
GRAPHIC MODE	ą	ć	ę	ł	ń	ó	ś	ż
GRAPHIC MODE + SYMBOL SHIFT	∃	δ	Θ	→	—	Σ	∫	■
SYMBOL SHIFT	I	π	#	$	%	&	'	(

```
10 CLEAR 64001: LET L=60: LET N=64002
20 LET S=0: FOR F=1 TO 50: READ A: LET S=S+A: POKE N,A: IF N=65367 THEN GO TO 40
30 LET N=N+1: NEXT F
40 READ A: IF A<>S THEN PRINT AT 10,7; FLASH 1;"POPRAWIC LINIE ";L: STOP
50 LET L=L+10: IF N<65367 THEN GO TO 20
60 DATA 17,10,250,175,205,13,12,201,128,127,65,46,90,97,103,114,111,100,122,10
5,110,115,107,233,58,58,92,254,11,40,33,253,203,1,126,32,7,33,26,250,229,195,183
,18,205,3,19,253,54,0,5292
70 DATA 255,42,89,92,205,167,17,33,26,250,229,195,180,18,42,93,92,43,126,254,1
73,32,214,253,54,0,255,62,2,205,48,37,196,1,22,205,77,13,205,133,250,253,203,0,1
26,40,190,223,254,13,6187
80 DATA 40,8,253,54,0,11,254,58,32,177,253,54,0,255,32,26,250,229,205,48,37,19
4,118,27,33,183,18,229,195,118,27,223,205,69,32,40,13,205,164,251,40,251,205,166
,250,205,164,251,40,243,6436
90 DATA 254,41,200,205,195,31,62,13,215,253,203,2,182,201,222,254,172,32,17,20
5,121,28,205,195,31,205,7,35,120,205,221,250,62,22,24,27,254,173,32,46,221,205,1
30,28,62,32,205,35,251,205,6607
100 DATA 195,31,205,153,30,121,205,221,250,203,25,62,23,215,121,215,120,215,20
1,203,71,32,7,253,203,2,182,203,56,201,253,203,2,246,24,247,205,242,33,208,205,1
12,32,208,205,251,36,205,195,31,7367
110 DATA 253,203,1,118,40,19,42,143,92,229,58,145,92,245,239,46,56,241,50,145,
92,225,34,143,92,205,241,43,120,177,11,200,26,19,205,35,251,24,245,197,213,254,3
2,56,120,254,144,48,116,254,6533
120 DATA 128,48,105,214,32,1,215,251,111,38,0,41,41,41,9,17,207,251,213,253,20
3,2,118,32,13,1,8,0,237,176,209,205,3,11,14,15,24,22,6,8,126,203,63,203,63,203,6
3,203,63,18,4731
130 DATA 19,35,16,242,209,205,3,11,14,240,6,8,213,255,26,161,182,253,203,87,86
,40,1,169,119,35,20,16,241,209,205,3,11,205,127,11,253,203,2,118,32,2,43,12,205,
229,10,58,60,92,5176
140 DATA 238,64,50,60,92,209,193,201,214,128,1,215,254,14,149,193,193,201,223,
254,59,40,29,254,44,32,19,205,40,37,40,29,62,32,205,75,251,253,203,2,18,[illegible],6,21
5,24,6,254,39,192,205,6211
150 DATA 155,250,231,205,69,32,32,1,193,191,201,15,239,79,79,79,79,239,15,0,0,
0,0,0,0,0,0,0,64,64,64,64,0,64,0,0,160,160,0,0,0,0,0,0,0,160,224,160,224,160,395
2
160 DATA 0,64,96,128,224,32,160,64,64,128,32,32,64,64,128,128,32,32,80,32,96,1
60,144,240,0,0,128,128,0,0,0,0,0,0,64,128,128,128,128,64,0,0,64,32,32,32,32,64,0
,0,3376
170 DATA 0,0,160,64,160,0,0,0,0,0,64,224,64,0,0,0,0,0,0,0,0,32,64,0,0,0,0,224,
0,0,0,0,0,0,0,0,0,64,0,0,32,32,64,64,128,128,0,0,224,160,1952
180 DATA 160,160,160,224,0,0,64,192,64,64,64,224,0,0,224,160,32,64,128,224,0,0
,224,32,96,32,160,224,0,0,128,128,160,224,32,32,0,0,224,128,224,32,32,224,0,0,22
4,128,128,224,5248
190 DATA 160,224,0,0,224,32,32,64,64,64,0,0,224,160,224,160,160,224,0,0,224,16
0,224,32,160,224,0,0,0,0,64,0,0,64,0,0,0,0,64,0,0,64,128,0,0,32,64,128,64,32,374
4
200 DATA 0,0,0,0,224,0,224,0,0,0,0,128,64,32,64,128,0,0,64,160,32,64,0,64,0,0,
240,160,160,160,160,160,0,0,64,160,160,224,160,160,0,0,192,160,192,160,160,192,0
,0,4272
210 DATA 64,160,128,128,160,64,0,0,192,160,160,160,160,192,0,0,224,128,192,128
,128,224,0,0,224,128,192,128,128,128,0,0,64,160,128,224,160,64,0,0,160,160,224,1
60,160,160,0,0,224,64,5792
220 DATA 64,64,64,224,0,0,32,32,32,32,160,64,0,0,160,160,192,160,160,160,0,0,1
28,128,128,128,128,224,0,0,160,224,224,160,160,160,0,0,192,160,160,160,160,160,0
,0,64,160,160,160,5248
230 DATA 160,64,0,0,224,160,224,128,128,128,0,0,224,160,160,160,160,224,32,0,2
24,160,160,192,160,160,0,0,96,128,64,32,32,192,0,0,224,64,64,64,64,64,0,0,160,16
0,160,160,160,224,5504
240 DATA 0,0,160,160,160,160,160,64,0,0,160,160,160,224,224,160,0,0,160,160,64
,64,160,160,0,0,160,160,160,64,64,64,0,0,224,32,64,64,128,224,0,0,192,128,128,12
8,128,192,0,0,5024
250 DATA 128,128,64,64,32,32,0,0,96,32,32,32,32,96,0,0,64,224,64,64,64,64,0,0,
0,0,0,0,0,0,240,0,64,160,128,192,128,224,0,0,0,192,32,224,160,224,0,0,128,192,36
00
260 DATA 160,160,160,224,0,0,0,96,128,128,128,96,0,0,32,96,160,160,160,224,0,0
,0,96,160,192,128,96,0,0,32,64,96,64,64,64,0,0,0,96,160,160,96,32,192,0,128,192,
160,160,4544
270 DATA 160,160,0,0,64,0,192,64,64,224,0,0,32,0,32,32,32,160,64,0,128,160,160
,192,160,160,0,0,128,128,128,128,160,64,0,0,0,160,224,224,160,160,0,0,0,192,160,
160,160,160,4736
280 DATA 0,0,0,64,160,160,160,64,0,0,0,192,160,160,192,128,128,0,0,224,160,160
,160,[illegible]
0,[illegible]
290 DATA 0,160,160,160,160,224,0,0,0,160,160,160,160,64,0,0,0,160,160,224,224
[illegible]
[illegible]
300 DATA [illegible],64
[illegible],224
[illegible]
310 DATA 160,[illegible],96,128,128,128,96,0,0,0,96,160,192,128,96,64,0,128,
[illegible],160,64,[illegible],160,0,32,64,64,160,160,160,64,0,128,6
[illegible]
320 DATA 0,32,64,64,64,64,64,64,128,240,128,64,32,64,128,240,0,0,0,0,0,240,0,0
,0,0,0,0,32,240,32,0,0,96,144,208,176,144,144,96,0,0,192,32,96,160,160,64,0,0,36
[illegible]
330 DATA [illegible],240,240,240,[illegible]
```

SPECTRUM I DRUKARKA — DZM-180 —

Jednym z podstawowych urządzeń peryferyjnych komputera jest drukarka. Z uwagi na duże rozpowszechnienie w kraju mikrokomputerów **SINCLAIR ZX SPECTRUM** oraz drukarek **DZM-180** produkowanych przez **MERA-BŁONIE**, celowym stało się opracowanie urządzenia sprzęgającego (interface'u), umożliwiającego ich wzajemną współpracę.

W urządzeniu, którego schemat ideowy przedstawiono na rys. 1 zastosowano programowalny układ scalony **8255** firmy **INTEL** * (może być polski **MCY 7855** lub radziecki **KP 580 UK 55**), zawierający rejestr sterujący, bufor magistrali danych, trzy porty WEJŚCIE — WYJŚCIE **A**, **B**, **C** i połączenia wewnętrzne między nimi, dwa układy scalone **UCY 74 LS 04**, zawierające łącznie 12 bramek **NOT** i układ scalony **UCY 74 LS 02**, zawierający 4 bramki **NOR**.

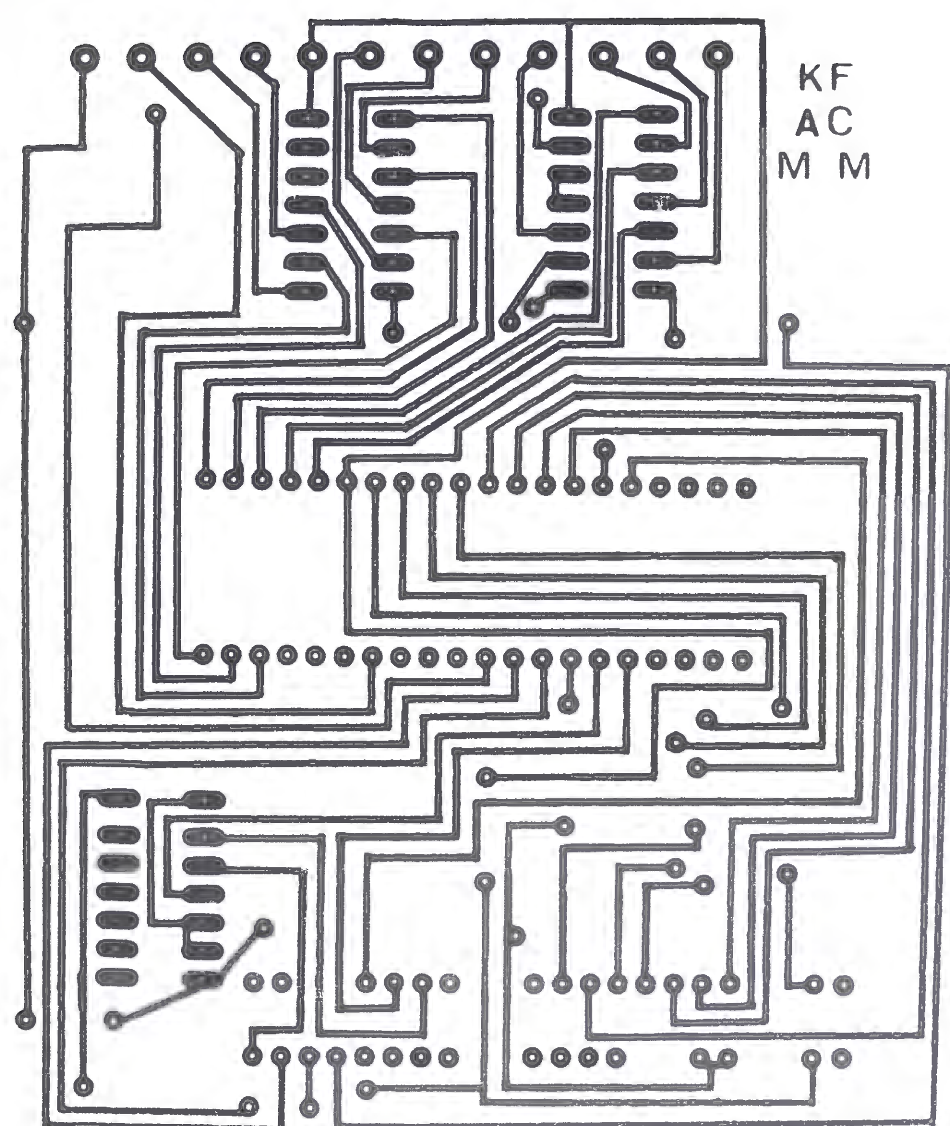
Rys. 1 Schemat ideowy interface'u

Do transmisji danych wykorzystano port **B** i dwie linie portu **C**, pozostałe linie portu **C** i port **A** można użyć do innych celów. Jako bity adresowe wykorzystano sygnały **A5**, **A6**, **A7** mikrokomputera. W celu zaprogramowania układu scalonego **8255** przyjęto słowo sterujące, które w formie decymalnej ma postać liczby 128. Ponadto dla ustalonego układu połączeń między mikrokomputerem, interface'm i drukarką przyjęto adresy poszczególnych portów i rejestru sterującego układu scalonego **8255** — które w formie decymalnej mają postać liczb — przy czym dla rejestru sterującego przyjęto adres w postaci liczby 223, dla portu **A** — w postaci liczby 31, dla portu **B** — w postaci liczby 95, dla portu **C** — w postaci liczby 159.

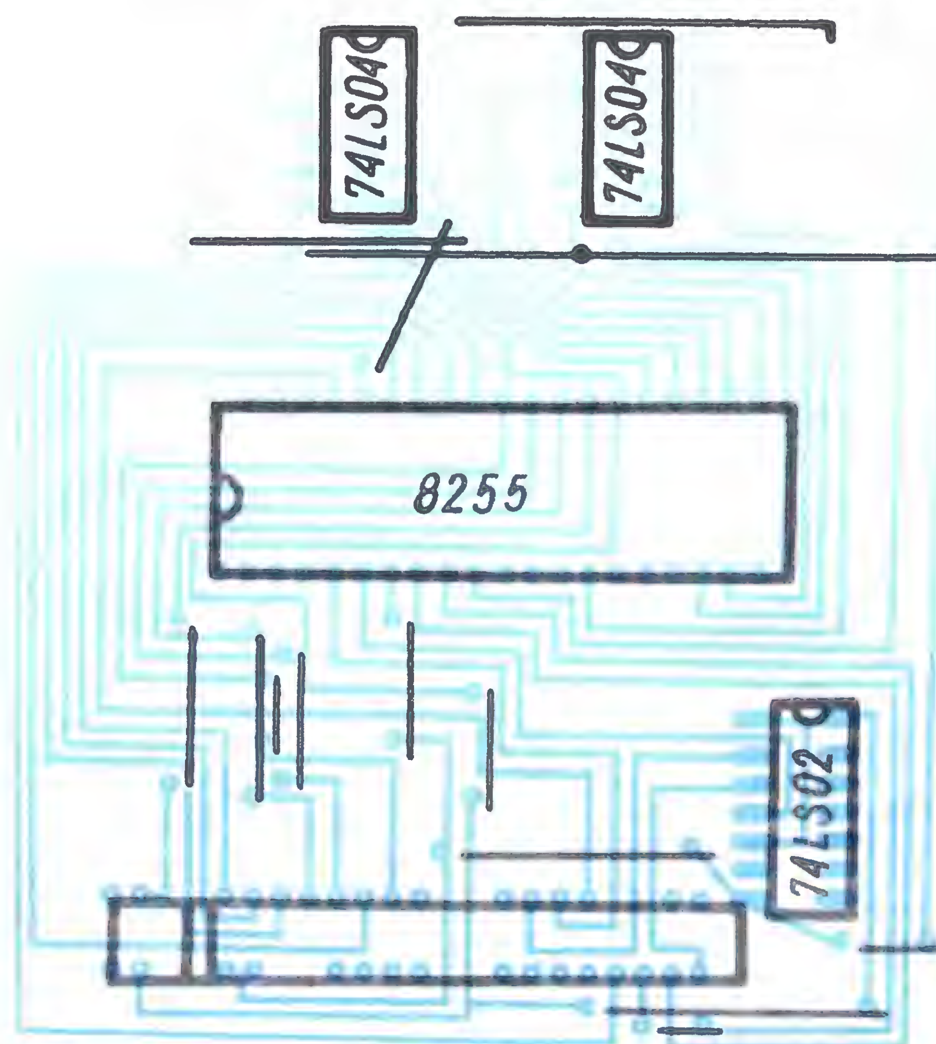
Między wyjście mikrokomputera i wejście rejestru sterującego układu scalonego **8255** w torze sygnału zerującego **Reset** włączona jest bramka **NOT** z układu scalonego **UCY 74 LS 04**, a w torze sygnału **CHIP SELECT** (cs) uaktywniającego układ scalony **8255**, między wyjścia **iorq** i **A7** mikrokomputera i wejście cs rejestru sterującego włączone są dwie połączone ze sobą szeregowo bramki **NOR** z układu scalonego **UCY 74 LS 02**. Między wyjścia portów **B** i **C** układu scalonego **8255** a wejścia układu sterowania i bufora drukarki w każdym z torów włączona jest bramka **NOT** z układu scalonego **UCY 74 LS 04**, a w torze sygnału **PC1-A6**, wpisującego informację do bufora drukarki dołączona jest druga bramka **NOT**.

Układ zmontowano na płytce drukowanej, przedstawionej na rys. 2. Do połączenia interface'u i mikrokomputera wykorzystano złącze krawędziowe **ZX SPECTRUM**. Schemat montażowy układu przedstawiono na rys. 3. Ponieważ drukarka **DZM-180** posiada jednobitową szynę danych, więc linia **PB7** portu **B** nie jest połączona. Schemat złącza krawędziowego dru-

* Dokładny opis budowy i programowania układu 8255 znajduje się w książce **Piotra MISIUREWICZA „Układy mikroprocesorowe"**



Rys. 2 Płytka drukowana interface'u skala 1:1



Rys. 3 Schemat montażowy, widok od strony elementów

karki **DZM-180** przedstawiono na rys. Połączenie wyjścia interface'u z wejściem drukarki należy wykonać możliwie najkrótszym dziesięciożyłowym przewodem zaopatrzonym w typowe złącze krawędziowe.

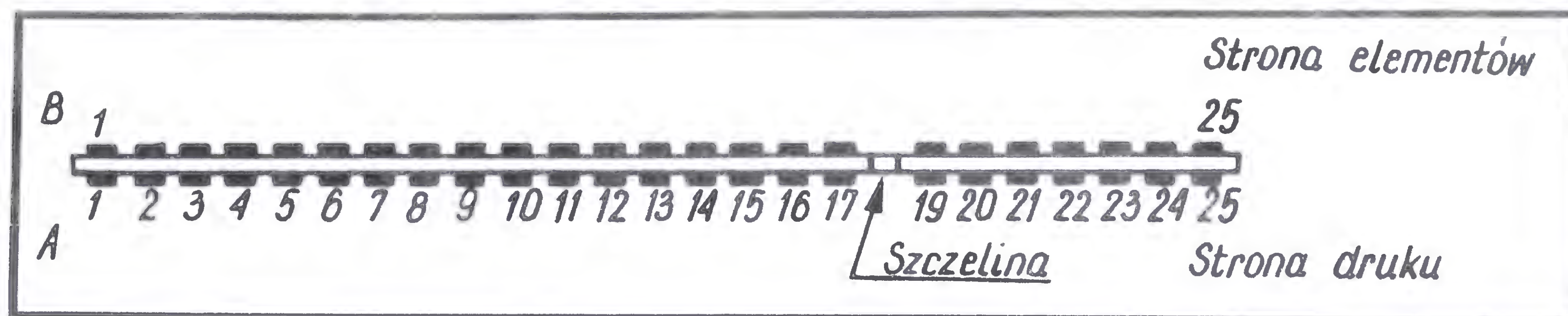
Do przesyłania informacji opracowano program obsługi interface'u w języku **BASIC**, składający się z podprogramów **LLIST** i **COPY**. Program obsługi należy dołączyć do programu użytkownika za pomocą instrukcji **„MERGE"**. Podprogramy uruchamia się instrukcją **GO TO 9000** dla **LLIST** i **GO TO 9300** dla **COPY**.

W celu uproszczenia programu obsługi zrezygnowano z systemu przesyłania danych z potwierdzeniem. Bardziej zaawansowani w programowaniu mogą napisać program obsługi interface'u w języku wewnętrznym mikroprocesora z wykorzystaniem systemu przesyłania danych z potwierdzeniem.

Wykaz elementów

8255 (MCY 7855, KP 580 UK 55)	— 1 szt.
UCY 74 LS 04	— 2 szt.
UCY 74 LS 02	— 1 szt.
40 nóżkowa podstawka pod układ scalony	— 1 szt.
złącze krawędziowe do mikrokomputera	— 1 szt.
złącze krawędziowe do drukarki	— 1 szt.
10 żyłowy przewód łączący	— ok. 1,5 m

Andrzej Ciepliński
Konrad Fedyna
Marek Mizeracki



Schemat złącza krawędziowego drukarki DZM-180

```
9000 REM   PROGRAM OBSLUGI Nr.1
9001 REM       *   LLIST   *
9002 OUT 223,129: OUT 159,255: OUT 95,10: OUT 159,0:
 OUT 159,255
9003 LET r1=PEEK 23635+PEEK 23636*256: INPUT "NUMER
PIERWSZEJ LINII ? ";r7
9004 INPUT "NUMER OSTATNIEJ LINII ? ";r8: INPUT "ILO
SC ZNAKOW W WIERSZU ? ";dddg
9006 CLS : LET r2=r1+4: LET r3=(PEEK (r1+2)+256*PEEK
 (r1+3)
9007 LET r6=(256*PEEK r1)+PEEK (r1+1): IF r6=r7 THEN
 GO TO 9010
9009 LET r1=r2+r3: GO TO 9006
9010 LET r2=r1+4: LET r9=((256*PEEK r1)+PEEK (r1+1))
9020 IF r9>r8 THEN  GO TO 9200
9050 PRINT r9;: LET r3=(PEEK (r1+2))+256*PEEK (r1+3)
9060 FOR r=r2 TO r2+r3-1
9065 IF r=r2+r3-1 THEN  LET prh=33-PEEK 23688: LET p
rv=24-PEEK 23689: GO TO 9100
9070 IF PEEK r=14 THEN  LET r=r+5: GO TO 9090
9080 PRINT CHR$ (PEEK r);
9090 NEXT r
9100 LET r5=0: FOR i=0 TO prv-1: FOR j=0 TO 31
9125 LET r5=r5+1: IF r5=dddg THEN  GO SUB 9220
9130 LET r4=CODE SCREEN$ (i,j)
9140 IF r4<32 THEN  LET r4=32
9150 IF r4>97 AND r4<126 THEN  LET r4=r4-32
9155 IF r4>=127 AND r4<=164 THEN  LET r4=32
9160 OUT 95,r4: OUT 159,0: OUT 159,255
9170 NEXT j: NEXT i
9175 FOR j=0 TO prh: LET r4=CODE SCREEN$ (prv,j)
9180 LET r5=r5+1: IF r5=dddg THEN  GO SUB 9220
9181 IF r4<32 THEN  LET r4=32
9182 IF r4>=97 AND r4<=126 THEN  LET r4=r4-32
9183 IF r4>=127 AND r4<=164 THEN  LET r4=32
9184 OUT 95,r4: OUT 159,0: OUT 159,255
9185 NEXT j
9190 CLS : LET r1=r2+r3
9195 OUT 159,255: OUT 95,10: OUT 159,0: OUT 159,255:
 GO TO 9010
9200 STOP
9220 OUT 159,255: OUT 95,10: OUT 159,0: OUT 159,255
9230 LET r5=0: FOR n=1 TO 5
9250 OUT 159,255: OUT 95,32: OUT 159,0: OUT 159,255:
 LET r5=r5+1: NEXT n: RETURN
9251
9300 REM   PROGRAM OBSLUGI Nr.2
9301 REM       *   COPY   *
9320 OUT 159,255: OUT 223,128: OUT 159,0: OUT 159,25
5: FOR i=0 TO 21: FOR j=0 TO 31
9330 LET r4=CODE SCREEN$ (i,j)
9335 IF r4>=97 AND r4<=126 THEN  LET r4=r4-32
9340 IF r4>=127 AND r4<=164 THEN  LET r4=32
9350 OUT 159,255: OUT 95,r4: OUT 159,0: OUT 159,255
9360 IF j=31 THEN  OUT 159,255: OUT 95,10: OUT 159,0
: OUT 159,255
9380 NEXT j: NEXT i: STOP
```

TRZY TYGODNIE W RAJU

CZYLI PRZYGODY WALLY'EGO NA WYSPIE LUDOŻERCÓW

Wally, znany już z programów „Pyjamarama" i „Everyones Wally", udał się tym razem w podróż z Ameryki do Europy. Los zrządził, iż statek rozbił się, a wyrzuceni na ląd: Wally, jego żona Wilma i syn Hubert wpadli w łapy okrutnych ludożerców. Wally zdołał wyrwać się z niewoli, ale Herbert i Wilma stanowić mają główny przysmak ludożerców na uczcie, wydawanej na cześć Bożka Deszczu. Wally, w którego postać wciela się grający, ma do wykonania dwa zadania: uwolnić rodzinę i zbudować tratwę, na której wszyscy będą mogli bezpiecznie opuścić niegościnną wyspę.

Grając budujesz tratwę, każde prawidłowo wykonane „zadanie" to jej jeden segment. Kolejność wykonywania zadań jest do pewnego stopnia dowolna. Ja podaję dosyć szybki, mało męczący i chyba bardziej optymalny wariant.

1. UWOLNIENIE HERBERTA

Herbert jest gotowany w wielkim kotle. Pilnują go dwa wielkie LWY. Uwaga... nie podchodź bo ugryzą...

— Zauważ, że prawy LEW ma ogon przybity kołkiem do ziemi — pomóż mu.

Aby uwolnić LWA należy:

1. Zabrać z poczty (Indian Post) KLAPKI (Flip-Flops).
2. Znaleźć WIADERKO (CAN).
3. Pójść z tymi rzeczami do Gejzerka tzn. przejść przez „Kuźnię", „Pokój Jadalny", po schodach w prawo, przez polanę z domkiem i samochodem do polany z „Gejzerkiem".
4. Mając KLAPKI (Flip-Flops) i WIADERKO (CAN) pociągnąć za sznurek i szybko dojść do „Gejzerka", wtedy napełnisz WIADERKO (EMPTY CAN-FULL CAN).
5. Przejść do „Gejzerka" na plażę, gdzie siedzi KRAB — UWAGA! ... musisz mieć KLAPKI ze sobą.
6. Użyć „FUL CAN" na Kraba (klawisz np. ENTER), wtedy odpadną mu SZCZYPCE, które należy zabrać w miejsce po WIADERKU.
7. KLAPKI zostaw na brzegu tuż przed plażą — będą Ci jeszcze potrzebne.
8. Ze SZCZYPCAMI Kraba dojdź do polany, na której gotuje się HERBERT i przy prawym LWIE „użyj" go (klawiszem np. ENTER). LEW zacznie „machać" ogonem, będzie Ci sprzyjać.
9. Następnie musisz zabrać dwa skrzyżowane polana (DEUX STICK) sprzed paszczy Krokodyla i zanieść do „Kuźni"
10. Przejdź przez „Pokój Jadalny", dalej schodami w lewo aż do „Studni". Na pierwszy kafelek od lewej strony można wskoczyć.
11. Wejdź na studnię i zabierz MIECH (BELLOWE). UWAGA! ... nie wciskaj tylko np. ENTER, bo wpadniesz do studni.
12. Wróć do „Kuźni" i rozpal ogień tzn. mając MIECH i DREWNO po kolei użyj DREWNA do rozpalenia ognia i MIECHA do jego zgaszenia. Zdmuchując ogień otrzymasz POPIÓŁ.
13. Weź POPIÓŁ i MIECH i dojdź do Bożka Deszczu — stoi tuż przed pocztą.
14. Złóż w ofierze Bożkowi POPIÓŁ (klawisz ENTER). Bożek zacznie podskakiwać, a wisząca na niebie chmura przesunie się w prawo do palm i tam się zatrzyma.
15. Podejdź do chmury z jej prawej strony. Chmura zacznie „płynąć" w lewo — idź za nią. UWAGA! ... jeżeli wyprzedzisz chmurę to wróci ona do punktu wyjścia — chmura przesuwa się tylko wtedy, gdy masz MIECH. Gdy go wypuścisz chmura stanie, weźmiesz — ruszy.
16. Chmura przesunie się — powoli — przez trzy obrazki, aż do domku. Idź za nią — chmura rozbije domek piorunem, a na gruzach zostanie tylko MUSZLA. Wtedy zostaw MIECH i weź MUSZLĘ.
17. Idź z MUSZLĄ do „Studni" i wskocz do środka (klawisz ENTER IN).
18. Na dnie studni spadającą kroplami wodą napełnij MUSZLĘ. Zabierz BUTELKĘ i po prawej ścianie wyjdź (klawisz ENTER) przez trzy pionowe ściany ze studni.
19. Z „pełną" MUSZLĄ idź do LWÓW i kotła gdzie ludożercy gotują HERBERTA, a następnie „użyj" MUSZLI (klawisz np. ENTER).
20. Po „użyciu" MUSZLI HERBERT będzie URATOWANY (pokaże się z prawej strony, w miejsce małego szkielecika).

2. JAK URATOWAĆ WILMĘ?

WILMA jest zawieszona na drzewie pilnowanym przez strażnika. UWAGA! ... Nie podchodź zbyt blisko, strażnik jest niebezpieczny — musisz go zabić, aby uwolnić WILMĘ.

1. Najpierw trzeba naostrzyć TOPOR (AXE):

— uprzednio przyniesioną BUTELKĘ należy zanieść do „Krokodyla" i tam ją zostawić;

— poszukaj KORKOCIĄG w „Kuźni" i TOREBKĘ WILMY na „Plaży";

— obie rzeczy zanieś do „Krokodyla" (gdy masz TOREBKĘ Krokodyl nic Ci nie zrobi) i pojedyńczo (najpierw KORKOCIĄG potem BUTELKĘ) przenieś przed znajdujący się za Krokodylem KOKOS;

— nawierć KOKOS (klawisz np. ENTER) mając ze sobą BUTELKĘ. BUTELKA napełni się OLEJEM;

— wychodząc zabierz TOPÓR i BUTELKĘ z OLEJEM;

— obie rzeczy zanieś na polankę, gdzie stoi samochód z kwadratowymi kołami. Tam „użyj" tych przedmiotów (klawisz np. ENTER);

— TOPÓR zrobi się ostry (SHARP AXE).

2. Wracając do WILMY z TOPOREM, weź z „Jadalni" MISKĘ z KASZĄ i obie rzeczy zanieś do pilnującego WILMĘ strażnika. Zostaw TOPÓR.
3. Z MISKĄ PEŁNĄ KASZY przejdź (w lewo) do polany, na której stoi duży „Kurczak".
4. Gdy masz MISKĘ „Kurczak" zniesie JAJO.
5. Zabierz JAJO i zanieś je do „Gejzerka" i zostaw — jeszcze Ci się przyda.
6. Idź na „Pocztę" (Indian Post) i przed wejściem, tuż za słupem znajdziesz MINT (jest niewidoczny!).
7. Z MINTEM (moneta z małym otworkiem) idź do „Krokodyla", przejdź go „przy użyciu" TOREBKI i dojdź do następnej polany (Zamrożony Las), gdzie znajdziesz sześcienny KRYSZTAŁ.
8. Użyj MINTA przy KRYSZTALE (klawisz np. ENTER) a powstanie DZIURA (THE HOLE). Weź ją (UWAGA!.. nie będzie jej widać, tylko napis THE HOLE przypomni Ci o niej)
9. Z DZIURĄ idź do „Studni" zabierając po drodze DREWNIANĄ CZARODZIEJSKĄ KULĘ (BOWL).
10. Mając przy sobie KULĘ i DZIURĘ stań przy lewej ścianie „Studni". Gdy użyjesz DZIURY w wysokim murze powstanie wyrwa. Wejdź do środka.
11. Znajdziesz się w „Skarbcu". Zobaczysz szkielety śmiałków, którzy próbowali szczęścia przed Tobą. Nie bój się — chroni Cię MAGICZNA KULA. Gdybyś jej nie miał, pająk postarał by się o to, byś dołączył do poprzedników.
12. Zabierz jedynie KLUCZ (SKELETON KEY) i wyjdź. Przy studni zostaw KULĘ (nie będzie Ci już potrzebna).
13. Z KLUCZEM idź na plażę, weź KLAPKI (Flip-Flops) i przejdź na brzeg morza.
14. Zanurkuj.
15. Pod wodą otwórz „Szafkę Podmorską", zabierz z niej SZPINAK (SPINACH) w pudełku i wypłyń na powierzchnię — wróć do „Gejzerka".
16. Weź JAJO. Niosąc również SZPINAK pociągnij za sznurek i szybko podejdź do Gejzerka.
17. Wejdź do „działającego" Gejzerka, a on przeniesie Cię na szczyt drzewa, do „GNIAZDA ORŁA".
18. Z gniazda zabierz ŁUK I STRZAŁĘ i zostawiając SZPINAK, wyskocz z gniazda (przez strumień Gejzerka spadniesz na ziemię).
19. Na dole zostaw JAJO (nie będzie Ci już potrzebne).
20. Z ŁUKIEM I STRZAŁĄ pójdź do miejsca, gdzie wisi WILMA.
21. Gdy będziesz strzelał z ŁUKU (klawisz np. ENTER) do „Chodzącego Wodza" ten zmieni kierunek marszu.
22. Na polanie, gdzie uwięziona jest WILMA, strzel z ŁUKU do pilnującego WILMĘ strażnika. Strażnik zniknie.
23. Weź pozostawiony tu uprzednio TOPÓR i odetnij (klawisz ENTER) linę, na której wisi WILMA. Pokaże się w miejsce większego szkielecika.

Zauważ, że grając zbudowałeś już całą TRATWĘ — możesz uciekać z wyspy. Z całą rodziną idź na plażę. Wejdź do wody, a Wally odpłynie na tratwie wraz z rodziną na poszukiwanie nowych przygód.

Sterowanie WALLY'm:

— Drążkiem sterowym w zależności od Interface'u Kompston, Protek.

— z klawiatury:

= ruch w lewo — kl. od „Q" do „0" co drugi klawisz;

= ruch w prawo — kl. od „W" do „P" co drugi...

= skok — kl. od Caps do Symbol Shift + Space;

= wchodzenie IN przy drogowskazach IN, a także „używanie" niesionych przedmiotów — ENTER + kl. od „A" do „L";

ponadto:

= Klawisz „1" — zabieranie przedmiotów na miejsce pierwsze, górne, gdy nic nie niesiesz jest tam napis NOTHING;

= klawisz „2" — jak wyżej, przedmiot jest na miejscu drugim;

= klawisz „4" — STOP. Pauza w grze HOLD;

= klawisz „5" — włączenie i wyłączenie muzyki MUSIC ON/OFF.

Rafał Orlikowski

POKE rzysta

W prezentowanej w 3 numerze grze „Jet Set Willy" mamy możliwość ułatwienia sobie gry i to kilkoma sposobami.

Pierwszy z nich, to wprowadzenie instrukcji **POKE 35899 0**. Powoduje ona nieśmiertelność.

Niektórzy też pewnie wiedzą, jak trudno zdobyć skarby znajdujące się w kuchni. Sprawę ułatwiają:

POKE 60231,0 : POKE 42183,11 : POKE 56876,4 : POKE 59901,81

Dzięki nim, po spotkaniu z „kucharzami" nie traci się życia. Całkowitym ułatwieniem jest usunięcie z gry właściwie wszystkich stworów. Należy tylko wpisać te pętle:

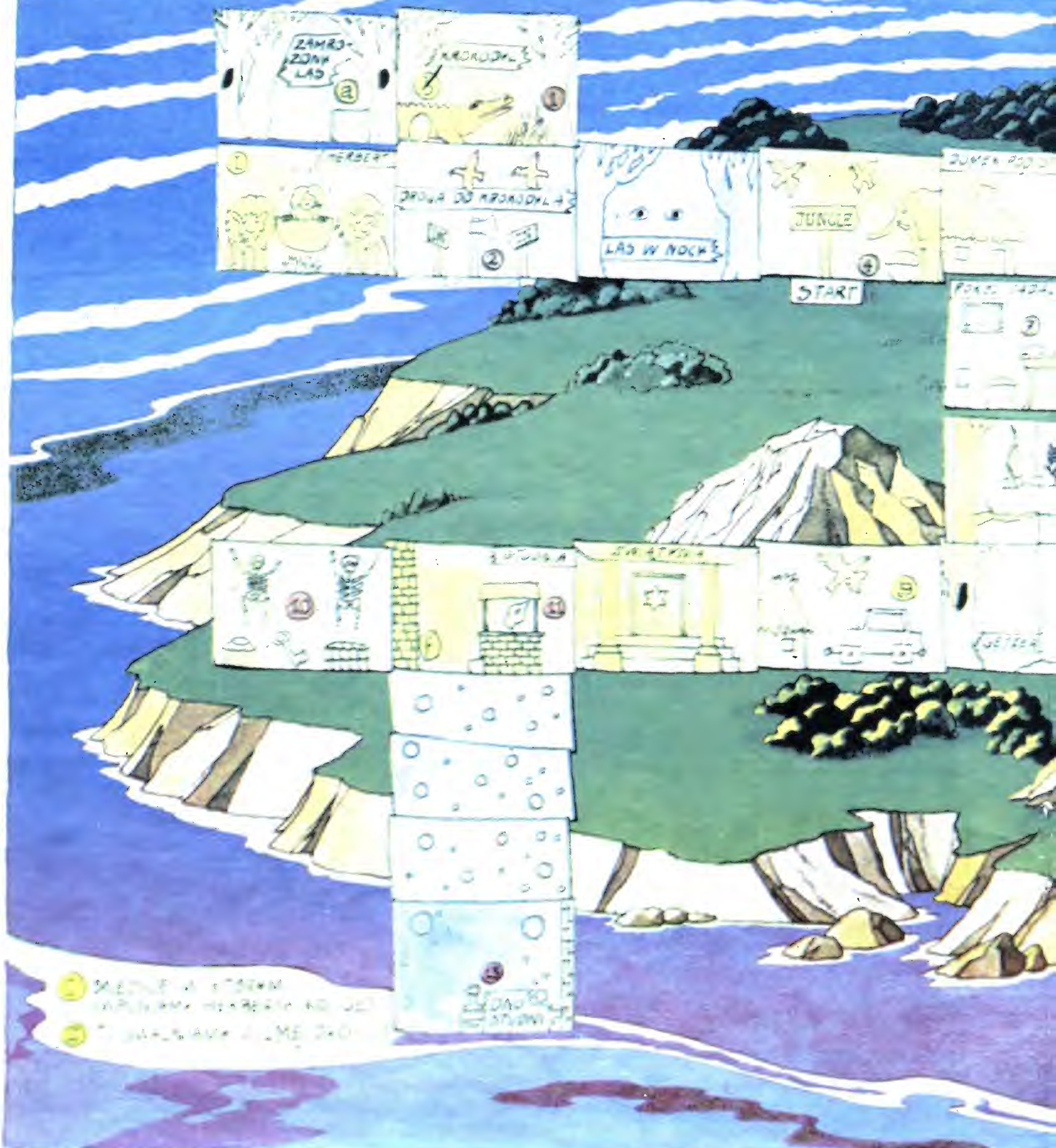
FOR f = 40000 TO 40191 : POKE f,0 : NEXT f :
FOR f = 43780 TO 46595 : POKE f,0 : NEXT f :
FOR f = 46845 TO 49178 : POKE f,0 = NEXT f

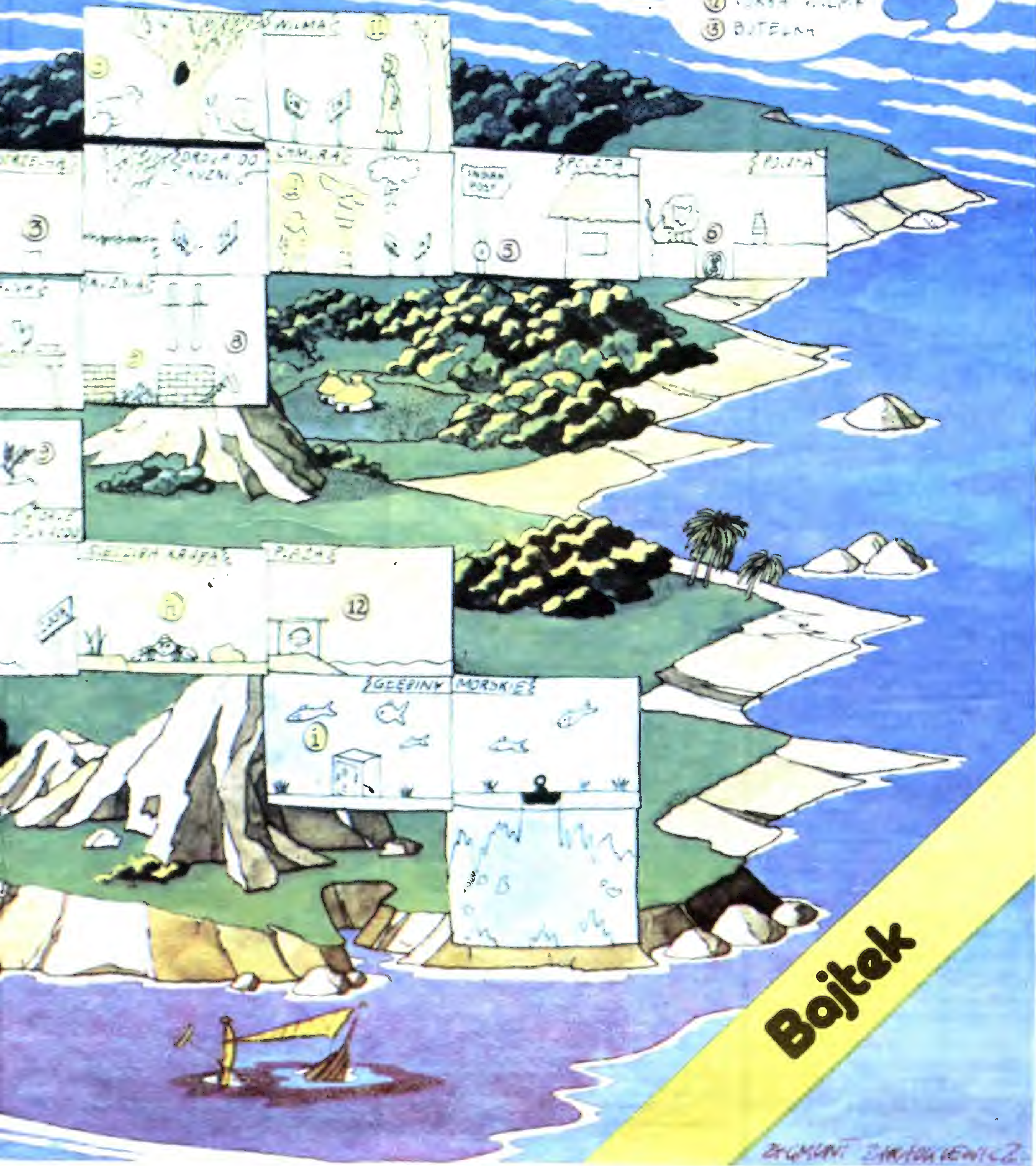
Stosowanie tego sposobu ma jednak swoją wadę. Po załadowaniu programu, należy poczekać ok. 30 sekund, gdyż muszą być wykonane pętle.

Oczywiście nie wpisuje się tego wszystkiego w dowolnym momencie. Po rozpoczęciu wgrywania programu ukazuje się najpierw nagłówek, a po chwili słychać krótką melodyjkę i widać napis „Jet Set Willy Loading". Wtedy trzeba wyłączyć magnetofon, wcisnąć **BREAK** z **CAPS SHIFT** i następnie **LIST**.

POKE'i i pętle można wpisać wszystkie razem, w linii 35, oddzielając je dwukropkami. Następnie wcisnąć trzeba **RUN** i włączyć magnetofon. Na wszelki wypadek można trochę cofnąć taśmę.

TRZY TYGODNIE W RAJU





PRZEDMIOTY DO NOSZENIA
TOPOR
MONETA Z DZIURKĄ
BUTELKA
KRAB
WILMA
GŁĘBINY MORSKIE
Bajtek



NOTOWANIE CZWARTE

Otrzymaliśmy 1648 propozycji do 4 notowania BAJTKOWEJ LISTY PRZEBOJÓW. Rekordową ilość głosów otrzymała gra KNIGHT LORE. Na I miejscu umieściło ją ponad 30% czytelników! A całość wygląda tak:

1 KNIGHT LORE ⇦
2 SUPERCHESS !
3 BOOTY !
4 UNDERWURLDE !
5 CYCLONE !
6 MUGSY !
7 THE WAY OF THE EXPLODING FIST ⇦
8 PYJAMARAMA !
9 MATCH POINT !
10 JUMPING JACK ⇦

Premie za najlepsze opisy typowanych gier przyznajemy: Maciejowi Chmielarzowi z Lublina oraz Arkadiuszowi Rajs z Bydgoszczy. Jest to bezpłatna prenumerata Bajtka na rok 1986/87.

Niestety tej nagradzanej dziesiątki nikt nie wytypował.

Sławek

Nasz adres:

BAJTEK
ul. Wspólna 61
00-687 Warszawa
LISTA PRZEBOJÓW

ROAD RACE

Jeśli ktoś jeszcze dzisiaj chce wydać grę samochodową musi wymyśleć coś więcej niż nudne jeżdżenie w kółko. Zrozumiała to firma Activision.

Great American Cross-Country Road Race to wyścig samochodowy w całej swej okazałości. Aby przejechać Stany Zjednoczone od zachodniego do wschodniego wybrzeża, trzeba pokonać 15 etapów, wybierając jedną z 4 tras. Gdy strzałka obrotomierza przekroczy liczbę 9000 obr/min musisz włączyć drugi bieg, jednak dopiero przy czwartym biegu możesz wyprzedzić konkurentów i narzucić swoje tempo.

Samochód reaguje bardzo dokładnie na poruszanie drążkiem sterowym. W oryginalny sposób opracowano dodawanie gazu: przyciśnięciem guzika ognia przyśpiesza się, a pchnięcie drążka do przodu włącza wyższy bieg. Jeżeli przegapi się włączenie biegu, silnik wyzionie ducha — idealny trening dla początkujących automobilistów. Uważajcie jednak, benzyna kończy się szybko i w każdej chwili możecie znaleźć się na poboczu.

Co 160 km na drodze rozstawione są stacje benzynowe. Możecie tu zatankować, naprawić uszkodzony przez rozgrzanie silnik, a nawet — czemu nie? — troszkę odpocząć. Cała droga usiana jest najrozmaitszymi przeszkodami: radarami, robotami drogowymi, nadjeżdżającymi z naprzeciwka samochodami. Dodatkowo zmienna pogoda i perspektywa prowadzenia pojazdu w nocy również nie ułatwiają zadania. Etapy zmieniają się dość często, ale nigdy się nie powtarzają. Jeżeli wyścig zakończy się z dobrym czasem, można wpisać swoje nazwisko na listę uczestników, a wynik zakodować na wieczność.



Chociaż Road Race nie posiada tak podniecającej grafiki jak Pitstop II, to jednak animacja jest bardzo udana. Gra przyciąga złożonością zadań.

(jz)

RESCUE ON FRACTALUS

Jest to gra dla prawdziwych mężczyzn. Wymaga podzielności uwagi a podobno jedynie mężczyźni są w stanie wykonywać kilka czynności na raz!

Sytuacja jest wyjątkowa i wymaga dużo odwagi — jesteś pilotem myśliwca Valkyrie, a Twoim zadaniem jest uratowanie życia pilotom samolotów zestrzelonych na planecie Fractalus. Cała planeta pokryta jest wysokimi górami, więc lecisz i nie wiesz co Cię za chwilę może spotkać. Obsługa dział przeciwlotniczych nie śpi! Znajdujący się po prawej stronie tablicy przyrządów, dalekosiężny wskaźnik LR sygnalizuje gdy znajdziesz się w pobliżu zestrzelonego samolotu. Zanim wyładujesz rozejrzyj się dobrze, ponieważ może być on już zajęty przez wroga. Jeśli wszystko jest w porządku na ekranie ujrzysz zbliżającego się pilota. Jednak zaczekaj lepiej aż zapuka, a dopiero potem otwórz drzwi, bo wrogowie, czyli czarne charaktery (choć w tym przypadku koloru zielonego) wchodzą bez pukania! Każdy uratowany pilot dostarcza Ci nowych sił, nowej energii nie mówiąc już o dodatkowych punktach. A energii potrzebujesz dużo, bo każde trafienie przez wroga lub zderzenie z górą bardzo Cię osłabia.



Zegary na tablicy przyrządów informują o wysokości lotu, zapasie paliwa, położeniu horyzontu, prędkości i kierunku lotu. Musisz cały czas pamiętać skąd przyleciałeś, aby móc tam wrócić i nabrać paliwa. Światła po lewej stronie tablicy sygnalizują czy powietrzny statek-matka jest w pobliżu. Liczby po prawej stronie tablicy wskazują ilość zestrzelonych jednostek wroga, ilość pozostałych do uratowania pilotów oraz ilość uratowanych ponad wyznaczone Ci zadanie.

A więc: obserwuj tablicę przyrządów, wypatruj wroga, strzelaj, omijaj góry i ratuj pilotów. Mężczyźni do dzieła! Mnie się to nie udało...

tb

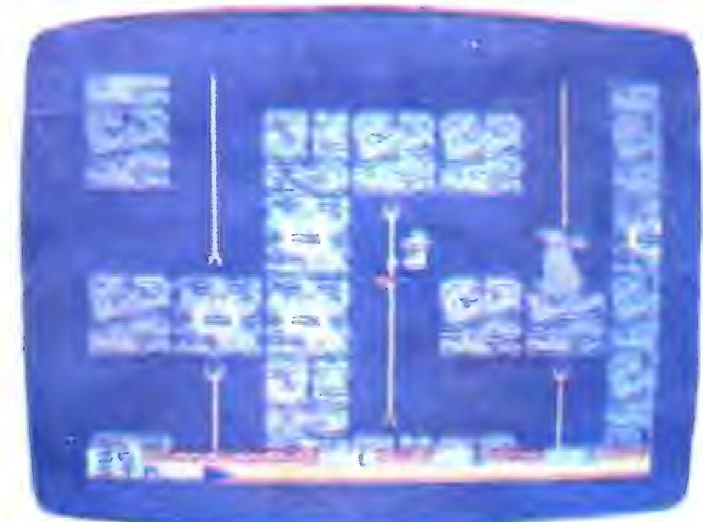
ROLAND ON THE ROPES

Znalazłeś się w ogromnym i dziwnym grobowcu. Jesteś na samym dnie i usiłujesz znaleźć drogę do wyjścia, które znajduje się w górze. W drodze ku wyjściu spotykasz wiele wrogo usposobionych stworów. Musisz być dzielny i nieustraszony, gdy wspinasz się po linach ku górze... Ale strzeż się! Każde spotkanie z potworami, kroplami kwasu, szczurami, mumiami, szkieletami, duchami i wampirami odbiera ci pewną porcję energii życiowej.

Aktualny zasób twojej energii wskazują strzałki u dołu ekranu, gdzie znajdziesz także inne informacje...

STR ◀◀◀◀◀ — zasób twoich sił SC — wynik gry L — poziom, na którym znajduje się S — nr labiryntu (do 4) B — ilość pocisków w twoim rewolwerze.

Jesteś nieustraszonym ROLANDEM' więc skaczesz ponad szczurami, by nie mogły cię dosięgnąć... Strzelasz do mumii, szkieletów i wampirów i nigdy nie chybiasz celu... Pamiętaj jednak, że strzelanie do duchów powstrzymuje je tylko przez chwilę, ale w zasadzie nie ma przed nimi ucieczki. Możesz najwyżej schować się w zakamarkach labiryntu grobowca.



Siły odzyskujesz zbierając amfory wypełnione eliksirem życia. Amunicja szybko wyczerpuje się warto więc zbierać butle z pociskami do rewolweru porzucone na korytarzach.

Musisz strzec się skorpionów, biegających szybko po ścianach pionowych szybów. Możesz je ominąć, odwracając się plecami ku ścianie, na której ich nie ma... Ale uważaj, bo potrafią one przeskoczyć ze ściany na ścianę... W nagrodę za zabicie potwora albo odnalezienie skarbu dostajesz pewną ilość punktów. Za wyjście z grobowca (ekran będzie wtedy radośnie błyskał) przewidziana jest premia.

tp

ZABEZPIECZENIA

Jeżeli chcesz zabezpieczyć Twój program przed zatrzymaniem go klawiszem **ESC** umieść w nim następującą instrukcję:

KEY DEF 66,0,0,0,0

Jeśli teraz naciśniesz ESC, SHIFT (ESC lub CTRL) ESC, to do komputera zostanie przesłany znak nr 0, który dla AMSTRADA oznacza „nic nie rób". Aby przywrócić stan poprzedni należy napisać:

KEY DEF 66,0,252,252,252

Możesz również zablokować kasowanie komputera (**CTRL**, **SHIFT** i **ESC**). W komórce pamięci o numerze **&BDEE** znajduje się początek procedury obsługi przerwania. Jeżeli wpiszesz do niej liczbę **& C9** oznaczającą w języku maszynowym powrót z procedury (tak, jak instrukcja **RETURN** w **BASIC-u**) to po każdym naciśnięciu klawiszy **CTRL**, **SHIFT** i **ESC** komputer natychmiast wróci do tego miejsca programu, w którym go przerwałeś. Wstawienie odpowiedniej liczby do komórki pamięci można uzyskać przy pomocy instrukcji **POKE:**

POKE & BDEE, & C9

Przed wykonaniem tego podstawienia powinno się zapamiętać poprzednią wartość:

KASOWANIE=PEEK (& BDEE)

aby można było do niej powrócić gdy będziesz chciał uaktywnić kasowanie komputera

Dariusz Wichniewicz



```
10 '*********************************
20 '*                               *
30 '*    ODTWARZANIE SKASOWANEGO    *
40 '*                               *
50 '*      PLIKU NA DYSKIETCE       *
60 '*                               *
70 '* autor: R.F.SPIEGEL   09.1985  *
80 '*                               *
90 '*********************************
100 MEMORY &7FFF:INK 0,0:MODE 2:INK 1,19
110 DATA &21,&37,&80,&cd,&d4,&bc,&22,&34,&80,&79,&32,&36,&80,&1e,&00,&16,&00,&0e,&00,&21,&39,&80,&df,&34,&80,&c9
120 DATA &21,&38,&80,&cd,&d4,&bc,&22,&34,&80,&79,&32,&36,&80,&1e,&00,&16,&00,&0e,&00,&21,&39,&80,&df,&34,&80,&c9
130 DATA &00,&00,&00,&84,&85
140 FOR i=&8000 TO &8038:READ a:POKE i,a:NEXT
150 FOR i=&8039 TO &8239:POKE i,&0:NEXT
160 nt=1:INPUT "JAKA NAZWA PLIKU ? ",nm$:l=LEN(nm$):IF l>8 THEN GOTO 160 ELSE IF l=0 THEN END ELSE nm$=UPPER$(nm$)
170 IF l<8 THEN nm$=nm$+" ":l=l+1:GOTO 170
180 INPUT "JAKI TYP PLIKU ? ",ty$:l=LEN(ty$):IF l<>3 AND l<>0 THEN GOTO 180
190 IF l=0 THEN nc=8 ELSE nc=11:nm$=nm$+UPPER$(ty$)
200 INPUT "KTORA STACJA DYSKOW (A/B) ? ",e$:e$=UPPER$(e$):IF e$<>"A" AND e$<>"B" THEN GOTO 200
210 IF e$="A" THEN POKE &800E,&0:POKE &8028,&0 ELSE POKE &800E,&1:POKE &8028,&1
220 INPUT "JAKI FORMAT DYSKIETKI (S=system, D=data, I=IBM) ? ",e$:e$=UPPER$(e$):IF e$<>"S" AND e$<>"D" AND e$<>"I" THEN GOTO 220
230 IF e$="S" THEN POKE &8010,&2:POKE &802A,&2:POKE &8012,&41:smax=&45:GOTO 260
240 IF e$="D" THEN POKE &8010,&0:POKE &802A,&0:POKE &8012,&C1:smax=&C5:GOTO 260
250 POKE &8010,&1:POKE &802A,&1:POKE &8012,&1:smax=&5
260 CALL &8000
270 ad=&8039
280 nsec=PEEK(&8012)
290 FOR i=1 TO 16:nf$=""
300 FOR j=0 TO nc-1:nf$=nf$+CHR$(PEEK(ad+j+1)):NEXT
310 IF nm$=nf$ AND PEEK(ad)=&E5 THEN nt=0:POKE ad,&0
320 ad=ad+32
330 NEXT i
340 POKE &802C,PEEK(&8012):CALL &801A
350 nsec=nsec+1:IF nsec<=smax THEN POKE &8012,nsec:POKE &802C,nsec:GOTO 260
360 IF nt=1 THEN PRINT "PLIK TAKI NIE ISTNIEJE":GOTO 160
365 PRINT "O.K. POZNIEJ MOZESZ SPRAWDZIC"
370 INPUT "SZUKASZ JESZCZE INNEGO PLIKU (T/N) ? ",e$:e$=UPPER$(e$):IF e$<>"T" AND e$<>"N" THEN GOTO 370
380 IF e$="T" THEN GOTO 160
390 END
```

CPC 6128

ODZYSKIWANIE PLIKÓW SKASOWANYCH OMYŁKOWO NA DYSKIETCE

Instrukcja |ERA w Basic Locomotiv umożliwia skasowanie pliku na dyskietce np. w celu uzyskania miejsca na wpisanie innego programu. Użycie tej samej instrukcji wymaga szczególnej ostrożności, ponieważ skasowany przy jej użyciu plik nie będzie wykazywany po użyciu instrukcji CAT (katalog).

Powyższy program napisany na BASIC-u umożliwia odtworzenie skasowanego przez omyłkę pliku, o ile po wykonaniu instrukcji |ERA na dyskietkę **nie został zapisany** inny plik.

WPROWADZENIE: Wykonanie instrukcji |ERA w odniesieniu do określonego pliku nie oznacza, że plik ten został wykasowany z dyskietki. AMSDOS zadowala się zbadaniem (w części katalogu dotyczącej tego pliku) wskaźnika stanu (flag) i stwierdzenia, czy plik ma być traktowany jako nieistniejący. Odczytanie skasowanego pliku polega więc na takiej modyfikacji wskaźnika stanu, aby system mógł go odczytać jako plik istniejący.

STRUKTURA I LOKALIZACJA KATALOGU NA DYSKIETCE: Usytuowanie katalogu zależy od formatu dyskietki.

— jeśli dyskietka sformatowana jest w typie SYSTEM to katalog znajduje się na ścieżce w 2 sektorach od 65 do 69 (hex. 41 do 45)
— jeśli dyskietka jest typu DATA to katalog znajduje się na ścieżce 0 między sektorami 193 i 197 (hex C1 do C5)
— jeżeli dyskietka ma format typu IBM to katalog umiejscowiony jest na ścieżce 1 w sektorach od 1 do 5.

Dla każdego pliku (a dokładniej dla każdego fragmentu pliku o długości maksymalnej 16k) katalog składa się z 32 bajtów. Nas interesuje głównie 12 pierwszych bajtów, zorganizowanych następująco:

— pierwszy bajt zawiera numer **USER**, oddziaływującego na plik. W tym bajcie znajduje się wskaźnik (flag) informujący o tym, czy plik jest skasowany czy nie. W przypadku, gdy plik jest skasowany, bajt ten przybiera wartość 229 (hex. E5),
— osiem następnych bajtów zapisanych jest nazwą pliku i ewentualnie uzupełnione spacjami,
— trzy następne bajty zawierają dane o rodzaju pliku (np. BAS, BIN lub inne).

MOŻLIWOŚCI SYSTEMU AMSDOS książka „DDI-1 Firmware" wydana przez firmę Amstrad opisuje programy istniejące w pamięci ROM stacji dysków DDI-1 dostępne dla systemu AMSDOS. Programy wywoływane są za pomocą instrukcji KL FIND COMMAND. Są one pomocne przy wyszukiwaniu instrukcji ulokowanych w RSX i w dodatkowej pamięci ROM. Ich nazwa zbudowana jest z jednego tylko znaku, którego ósmy bit powinien być jedynką, aby instrukcja ta była wykonalna.

Interesują nas dwie instrukcje:

— odczytywanie sektora z dyskietki (kod hex.04), który staje się dostępny przy zmianie kodu na hex.84 (ósmy bit staje się jedynką),
— zapisywanie sektora dyskietki (kod hex.05) możliwe po przyroście wartości do hex.85 (8 bit — jedynka)

Programy powyższe wywoływane są przy pomocy krótkiego programu w assemblerze, ładowanego programem napisanym w BASIC-u.

Przy korzystaniu z programu należy poda

— nazwę i typ pliku, który chce się reaktywować,
— format dyskietki, na której znajdował się ten plik oraz indeks stacji dysków (A lub B).

Program poszukuje w katalogu dyskietki pliku, którego nazwa i typ są identyczne z podanymi przez operatora i którego parametr **USER** równy jest hex.E5.

Za każdym razem, gdy dane katalogu odpowiadają wprowadzonej specyfikacji, jego numer **USER** ustawiany jest na 0. Jeżeli żaden z plików katalogu nie odpowiada podanej specyfikacji, na monitorze pojawia się komunikat **„PLIK TAKI NIE ISTNIEJE"** oraz **„PODAJ NAZWĘ PLIKU"**, po którym program oczekuje na wprowadzenie danych.

KORZYSTANIE Z PROGRAMU: Załadować program przy użyciu LOAD, włożyć do stacji dysków dyskietkę ze skasowanym omyłkowo programem (tą stroną dyskietki do góry, na której uprzednio znajdował się program skasowany), uruchomić program instrukcją RUN i wprowadzić informacje, których żąda program. Jeżeli plik został odtworzony, na ekranie monitora pojawi się komunikat **„O.K. POZNIEJ MOZESZ SPRAWDZIC"** i pytanie **„SZUKASZ JESZCZE INNEGO PLIKU (T/N)?"**. Odpowiedź „tak" (T) uruchamia program ponownie a „nie" (N) kończy pracę programu.

Na podstawie Amstrad Magazine

Wojciech Ziółek

TAJEMNICA MERITUM

Dla wszystkich użytkowników mikrokomputerów MERITUM mamy niespodziankę.

W tym artykule prezentujemy niektóre ich możliwości, o których nie wspomina producent. Należy do nich między innymi zestaw słów kluczowych **BASIC-a,** których znaczenie może samodzielnie zdefiniować użytkownik. W wersji dyskowej mikrokomputera MERITUM słowa te są wykorzystane przez producenta w rozszerzeniu **BASIC-a** wczytywanym z dyskietki systemowej. Użycie ich bez wcześniejszej deklaracji obsługi na pierwszej wersji MERITUM lub bez wprowadzenia rozszerzenia z dyskietki na wersji drugiej powoduje komunikat błędu ?L3 ERROR. Definiowanie odbywa się poprzez umieszczenie w pamięci RAM adresu procedury obsługi danego słowa. Do słów tych należą zarówno instrukcje jak i funkcje. Każdemu z nich przyporządkowane są dwie komórki w pamięci, w których umieścić należy adres procedury obsługi (kolejno młodszy i starszy bajt — starszy bajt=INT (adres/256), młodszy bajt= adres — starszy bajt * 256). Poniżej przedstawiamy listę słów kluczowych wraz z odpowiadającymi im adresami tych komórek.

INSTRUKCJE

CMD	16756	4174H	LINE	16804	41A4H	OPEN	16762	417AH
FIELD	16765	417DH	GET	16768	4180H	PUT	16771	4183H
CLOSE	16774	4186H	LOAD	16777	4189H	MERGE	16780	418CH
NAME	16783	418FH	KILL	16786	4192H	LSET	16792	4198H
RSET	16795	419BH	SAVE	16801	41A1H	DEF	16732	415CH

Po wywołaniu instrukcji w parze rejestrów HL zapisany jest adres tekstu programu następującego bezpośrednio po tym słowie. Można to wykorzystać do odczytu parametrów instrukcji, przy czym należy pamiętać o tym, że wracając do interpretora (przez RET) w HL musi znajdować się adres dalszej części tekstu programu (separatora przed następnym słowem kluczowym).

FUNKCJE ARGUMENTOWE

CVI	16723	4153H	CVS	16729	4159H	CVD	16735	415FH
EOF	16738	4162H	LOC	16741	4165H	LOF	16744	4168H
MKI$	16747	416BH	MKS$	16750	416EH	MKD$	16753	4171H

Argument funkcji można przekazać do własnego podprogramu obsługi wywołując procedurę o adresie 2687 A7FH, która ładuje do pary rejestrów HL wartość wyrażenia będącego argumentem. Wynik działania procedury można przekazać do **BASIC-a** poprzez parę rejestrów HL i kończąc swój program skokiem bezwarunkowym pod adres 2714 A9AH. W przypadku zakończenia przez RET wynik będzie równy parametrowi.

FUNKCJE BEZARGUMENTOWE

FN	16726	4156H	INSTR	16798	419EH	TIME$	16759	4177H
&	16789	4195H						

Po ich wywołaniu w parze rejestrów HL znajduje się adres miejsca programu, w którym znajduje się kod funkcji. Przy wracaniu do **BASIC-a** należy pamiętać, że w parze rejestrów HL musi znajdować się adres o 1 większy. Wynik można przekazywać korzystając z podprogramu o adresie 2714 A9AH

PRZYKŁADY

FIELD — rysowanie ramki

	dziesiętnie	szesnastkowo
PUSH HL	229	E5
LD DE,63	17,63,0	11,3F,0
LD HL,15360	33,0,60	21,0,3C
LD (HL),151	54,151	36,97
LD B,62	6,62	6,3E
INC HL	35	23
LD (HL),131	54,131	36,83
DJNZ-5	16,251	10,FB
INC HL	35	23
LD (HL),171	54,171	36,AB
LD B,14	6,14	6,E
INC HL	35	23
LD (HL),149	54,149	36,95
ADD HL,DE	25	19
LD (HL),170	54,170	36,AA
DJNZ-8	16,248	10,F8
LD B,62	6,62	6,3E
INC HL	35	23
LD (HL),181	54,181	36,B5
INC HL	35	23
LD (HL),176	54,176	36,B0
DJNZ-5	16,251	10,FB
INC HL	35	23
LD (HL),186	54,186	36,BA
POP HL	225	E1
RET	201	C9

KILL długość, wysokość — generowanie dźwięku
zakresy: od 0 do 255

	dziesiętnie	szesnastkowo
CALL 11036	205,28,43	CD,1C,2B
PUSH AF	245	F5
RST 8	207	CF
DEFB 44	44	2C
CALL 11036	205,28,43	CD,1C,2B
POP DE	209	D1
INC D	20	14
INC A	60	3C
LD E,A	95	5F
XOR A	175	AF
LD B,E	67	43
OUT (254),A	211,254	D3,FE
DJNZ-2	16,254	10,FE
DEC A	61	3D

JR NZ,-8	32,248	20,F8
DEC D	21	15
JR NZ,-11	32,245	20,F5
RET	201	C9

CVI (adres) — uruchamianie programu maszynowego od podanego adresu.

Uwaga: jeżeli adres okaże się większy od 32767, to należy jego wartość przekształcić wg wzoru: adres — 65536

	dziesiętnie	szesnastkowo
CALL 2687	205,127,10	CD,7F,A
JP (HL)	233	E9

INSTR — podawanie adresu pamięci, w której umieszczona jest dalsza część programu w Basicu

	dziesiętnie	szesnastkowo
INC HL	35	23
JP 2714	195,154,10	C3,9A,A

Program ładujący podane przykłady procedur może mieć następującą postać:

```
10 POKE 16765,48:POKE 16766,117:REM FIELD
20 DATA 229,17,63,0,33,0,60,54,151,6,62,35,54,131,16,
251,35,54,171,6,14,35,54,149,25,54,170,16, 248,6,62,35,
54,181,35,54,176,16,251,35,54,186,225, 201
30 POKE 16786,92:POKE 16787,117:REM KILL
40 DATA
   205,28,43,245,207,44,205,28,43,209,20,60,95,175,
   67,211,254,16,254,61,32,248,21,32,245,201
50 POKE 16723,118:POKE 16724,117:REM CVI (X)
60 DATA 205,127,10,233
70 POKE 16798,122:POKE 16799,117:REM INSTR
80 DATA 35,195,154,10
90 FOR A=30000 TO 30077:READ B:POKE A,B:NEXT
```

Po uruchomieniu programu ładującego można korzystać ze zdefiniowanych słów kluczowych, np. w krótkim programie wygrywającym melodię z filmu „Ojciec chrzestny".

```
100 CLS:PRINT @ 408,"OJCIEC CHRZEST-
    NY":FIELD:FOR A=1 TO 12:READ B: KILL
    0,B:NEXT
110 DATA
    239,179,150,159,179,150,179,159,179,226,201,239
120 A=CVI(73):PRINT @ 408,"ADRES KOŃCA PRO-
    GRAMU =";INSTR
```

We własnych programach maszynowych mogą okazać się przydatne następujące procedury:

73 49H — ładowanie do akumulatora kodu ASCII naciśniętego klawisza.

51 33H — wysyłanie na ekran znaku, którego kod ASCII zawarty jest w akumulatorze.

59 3BH — wysyłanie na drukarkę szeregową znaku j.w.

Wymienione procedury zmieniają zawartość pary rejestrów DE. Przy opracowywaniu własnych rozszerzeń Basica przydatna jest znajomość postaci, w jakiej napisany jest tekst programu.

Adres pierwszej linii programu zapisany jest w komórkach 16548 i 16549 (młodszy, starszy bajt). Postać zapisu linii: adres początku linii następnej (2 bajty), nr linii (2 bajty), tekst linii, kod zamykający =0.

Każde słowo kluczowe reprezentowane jest w postaci jednego bajtu:

128 80H END	129 81H FOR	130 82H RESET	131 83H SET
132 84H CLS	133 85H CMD	134 86H RANDOM	135 87H NEXT
136 88H DATA	137 89H INPUT	138 8AH DIM	139 8BH READ
140 8CH LET	141 8DH GO TO	142 8EH RUN	143 8FH IF
144 9OH RESTORE	145 91H GOSUB	146 92H RETURN	147 93H REM
148 94H STOP	149 95H ELSE	150 96H TRON	151 97H TROFF
152 98H DEFSTR	153 99H DEFINT	154 9AH DEFSNG	155 9BH DEFDBL
156 9CH LINE	157 9DH EDIT	158 9EH ERROR	159 9FH RESUME
160 AOH OUT	161 A1H ON	162 A2H OPEN	163 A3H FIELD
164 A4H GET	165 A5H PUT	166 A6H CLOSE	167 A7H LOAD
168 A8H MERGE	169 A9H NAME	170 AAH KILL	171 ABH LSET
172 ACH RSET	173 ADH SAVE	174 AEH SYSTEM	175 AFH LPRINT
176 BOH DEF	177 B1H POKE	178 B2H PRINT	179 B3H CONT
180 B4H LIST	181 B5H LLIST	182 B6H DELETE	183 B7H AUTO
184 B8H CLEAR	185 B9H CLOAD	186 BAH CSAVE	187 BBH NEW
188 BCH TAB	189 BDH TO	190 BEH FN	191 BFH USING
192 COH VARPTR	193 C1H USR	194 C2H ERL	195 C3H ERR
196 C4H STRING$	197 C5H INSTR	198 C6H POINT	199 C7H TIME$
200 C8H MEM	201 C9H INKEY$	202 CAH THEN	203 CBH NOT
204 CCH STEP	205 CDH +	206 CEH –	207 CFH x
208 DOH /	209 D1H [	210 D2H AND	211 D3H OR
212 D4H >	213 D5H =	214 D6H <	215 D7H SGN
216 D8 INT	217 D9H ABS	218 DAH FRE	219 DBH INP
220 DC POS	221 DDH SQR	222 DEH RND	223 DEH LOG
224 EOH EXP	225 E1H COS	226 E2H SIN	227 E3H TAN
228 E4H ATN	229 E5H PEEK	230 E6H CVI	231 E7H CVS
232 E8H CVD	233 E9H EOF	234 EAH LOC	235 EBH LOF
236 ECH MKI$	237 EDH MKS$	238 EEH MKD$	239 EFH CINT
240 FOH CSNG	241 F1H CDBL	242 F2H FIX	243 F3H LEN
244 F4H STR$	245 F5H VAL	246 F6H ASC	247 F7H CHR$
248 F8H LEFT$	249 F9H RIGHT$	250 FAH MID$	251 FBH '

Zmiana kodu reprezentującego dane słowo kluczowe (np. przez POKE) powoduje również zamianę w tekście programu tego słowa przez inne, którego reprezentacją jest nowo wstawiony kod. Umożliwia to samoprzekształcanie się programu np. w celu uzyskania wielowariantowości, poprawiania niektórych błędów itp. Dla użytkowników zainteresowanych utajnieniem treści programów w Basicu istnieje możliwość wykorzystania w tym celu edytora. Gotową linię należy zakończyć słowem REM. Przy pomocy edytora należy przejść do trybu dopisywania na końcu linii (klawisz X), klawiszami SHIFT i ↔ cofnąć kursor do początku ukrywanego tekstu, a następnie wpisać dowolny tekst. W mikrokomputerach MERITUM kody ASCII powyżej 192 powodują wygenerowanie łańcucha składającego się ze spacji i ilości kod-192.

W programach maszynowych i w Basicu przy pomocy PEEK i POKE korzystać można ze zmiennych systemowych interpretera, których skróconą listę zamieszczamy poniżej wraz z adresami zawierających je komórek:

16537 — kod ostatnio wciśniętego klawisza

16220 i 16221 — nr ostatnio wpisywanej lub poprawianej linii

16546 i 16547 — nr ostatnio wykonywanej linii

16561 i 16562 — wartość deklaracji MEMORY SIZE (OBSZAR PAO) — 2

16618 — 16619 — nr linii błędnej

16538 — wartość funkcji ERR (2 x nr błędu — 2)

16424 — ilość wierszy strony dla drukarki

16425 — ilość zapisanych wierszy strony dla drukarki

16416 i 16417 — adres komórki bufora ekranu, w której aktualnie znajduje się kursor

16551 i 16552 — adres początku bufora dla tekstów

Dla wszystkich liczb dwubajtowych w pierwszej komórce wpisywany jest młodszy, a w drugiej starszy bajt.

We własnych programach (BASIC i ASEMBLER) można odczytywać klawiaturę za pośrednictwem ustawianych sprzętowo komórek. Poniższe zestawienie przedstawia sposób przyporządkowania tych komórek poszczególnym klawiszom. Naciśnięcie klawisza powoduje ustawienie odpowiedniego bitu na 1.

Cezary Krajdener
Edmund Fliski

0	1	2	3	4	5	6	7	BIT	ADRES
@	A	B	C	D	E	F	G	14337	3801H
H	I	J	K	L	M	N	O	14338	3802H
P	Q	R	S	T	U	V	W	14340	3804H
X	Y	Z						14344	3808H
0	1	2	3	4	5	6	7	14352	3810H
8	9	:	;	,	-	.	/	14368	3820H
ENTER	CLEAR	BREAK	↑	↓	←	→	SPACE	14400	3840H
SHIFT	CTRL	▢						14464	3880H

ROZSZERZENIE MOŻLIWOŚCI KLAWIATURY

*Poniższy program przeznaczony na **MERITUM** i **MERITUM 1** wersja 2 rozszerza obsługę klawiatury o trzy OPCJE:*

1 — wciśnięcie dowolnego klawisza sygnalizowane jest dźwiękiem z głośnika w monitorze. Wysokość i długość dźwięku można zmienić poprzez zawartość komórek 30976 i 30974.

2 — dowolny znak będzie wczytywany z klawiatury dopóty, dopóki wciśnięty będzie odpowiadający mu klawisz.

4 — klawisz BREAK będzie ignorowany.

Każda z OPCJI działa niezależnie od pozostałych.

Wyboru OPCJI dokonujemy umieszczając w komórce 30953 numer wybranej opcji. Jeśli wybraliśmy kilka opcji, wówczas w komórce tej umieszczamy sumę ich numerów np.

POKE 30953,3 włącza opcję 1 i 2, zaś

POKE 30953,0 wyłącza wszystkie opcje.

A oto program:

```
1 FOR A = 30947 TO 31006 :
READ B : POKE A,B : NEXT
2 POKE 16407,120
3 DATA 205,227,3,183,200,1,0,3,
33,25
4 DATA 121,203,25,48,5,94,35,86,
43,213
5 DATA 35,35,16,243,201,8,62,50
6,100
6 DATA 211,254,16,254,61,32,
247,8,201,33
7 DATA 54,64,1,0,7,113,35,16,
252,201
8 DATA 61,200,60,201,252,120,
10,121,21,121
```

Program ten służy do załadowania procedury maszynowej uzupełniającej obsługę klawiatury. Może on być dołączony do większego programu lub po wykonaniu go skasowany przez NEW. Opcje 2 i 4 powinny być wyłączone przed zakończeniem programu. Ponadto opcja 2 mogąca utrudnić wpisywanie tekstu powinna być wyłączona przed INPUT. Dźwięk używany w opcji 1 może być wywoływany funkcją USR (X), jeśli do programu dołączymy linię.

```
9 POKE 16526,252 : POKE
16527,120
```

*Po użyciu przycisku **RESET** możemy wznowić działanie rozszerzeń obsługi klawiatury poprzez wykonanie sekwencji z linii nr 2, uprzednio dokonawszy wyboru opcji.*

Jeśli chcemy uniemożliwić przerwanie wykonywania programu należy oprócz BREAK zablokować również NMI sekwencją:

OUT 251,176

*Odblokowania **NMI** dokonujemy przez:*

OUT 250,255 : OUT 250,255

Radzę jednak przy stosowaniu blokad umieścić w programie możliwość przerwania programu przez STOP np. stosując hasło.

Życzę ciekawych pomysłów

Edmund Fliski



MIKROKOMPUTER NIE DO ZABAWY

NEW BRAIN

Wszystkim entuzjastom ZX SPECTRUM chciałbym dziś zaproponować zawarcie bliższej znajomości z jego rówieśnikiem o nieco poważniejszych aspiracjach. NEW BRAIN (Nowy Mózg) — bo tak właśnie nazywa się nasz nowy znajomy został zaprojektowany przez firmę Newbury Labs w roku 1981. Produkcję podjęła firma Grundy Business System i pierwsze egzemplarze pojawiły się na rynku wiosną 1982 roku. Półtora roku później właścicielem projektu stała się holenderska firma Tradecom International, która do dziś zajmuje się produkcją i dystrybucją NEW BRAIN'a.

Początkowo New Brain został pomyślany jako przenośny mikrokomputer do zastosowań biurowych, laboratoryjnych i edukacyjnych. W związku z tym zaprojektowano go tak, aby przy potencjalnie dużej pojemności informacyjnej ekranu (30 linii po 80 znaków) pozwalał elastycznie gospodarować swoimi zasobami (np. mieszanie tekstu i grafiki w różnych proporcjach) jak również umożliwiał łatwe komunikowanie się z urządzeniami zewnętrznymi.

W rezultacie powstał oryginalny produkt, o cechach niespotykanych wśród 8 bitowych mikrokomputerów osobistych. W chwili ukończenia projektu był pierwszym kandydatem do nagrody w konkursie BBC na mikrokomputer szkolny (konkurenci twierdzili nawet, że najpierw zaprojektowano New Brain'a a potem powstał regulamin konkursu!). Trudności finansowe firmy GBS sprawiły jednak, że nie został zwycięzcą. Obecni NEW BRAIN, mimo, że jest nadal produkowany w Holandii, zniknął prawie zupełnie z rynku brytyjskiego. Natomiast firma Tradecom International rozważa możliwość zbudowania montowni NEW BRAIN'a w Indiach i obiecuje nowy model z rozszerzoną do 128 kB pamięcią RAM.

BUDOWA

NEW BRAIN został zbudowany w oparciu o dwa mikroprocesory — popularny Z80A oraz specjalizowany 4 bitowy COP 420. Pierwszy z nich jest głównym procesorem systemu, natomiast drugi jest wykorzystywany do obsługi urządzeń peryferyjnych. W podstawowej wersji NEW BRAIN zawiera 32 kB pamięci RAM i 29 kB pamięci ROM. Pamięć RAM umieszczono w adresach od =0000 do =7FFF dzięki czemu na NEW BRAIN'ie można z łatwością zaimplementować CP/M — najpopularniejszy dyskowy system operacyjny dla mikrokomputerów 8 bitowych.

W odróżnieniu od innych mikrokomputerów. NEW BRAIN'a zbudowano bez wykorzystania specjalizowanych układów wysokiej skali integracji. Brak w nim odpowiednika układu ULA. Mimo, że rozmiarami niewiele większy od ZX Spectrum, został wyposażony w mechaniczną, bardzo trwałą klawiaturę o standardowych rozmiarach. (Jedyne odstępstwo to krótszy niż zazwyczaj klawisz spacji).

Bogatszy z dwóch produkowanych modeli — NEW BRAIN AD ma wbudowany 16 znakowy alfanumeryczny wyświetlacz fluorescencyjny, którym można posługiwać się niezależnie od ekranu. Do komputera można podłączyć odbiornik TV lub monitor. W trybie znakowym można wyświetlić od 1 do 30 linii tekstu po 40 lub 80 znaków każda. Wbudowany na stałe zestaw znaków jest bardzo bogaty. Obejmuje cały alfabet łaciński, specjalne znaki alfabetów zachodnioeuropejskich oraz, uwaga! pełny alfabet grecki (duże i małe litery). Ponadto znaki specjalne do realizacji prostej grafiki. W sumie jest 512 różnych znaków, jednak jednocześnie na ekranie można wyświetlić tylko 255 z nich.

W trybie graficznym w jednej linii można wyświetlić 256, 320, 512 lub 640 punktów. Linii na ekranie może być od 10 do 250. (A zatem rozdzielczość grafiki nie gorsza niż w IBM PC!). Przy tym górna część ekranu nie zajęta przez grafikę może być nadal używana do wyświetlania tekstów w trybie znakowym.

Podstawowym rodzajem pamięci masowej dla NEW BRAIN'a jest,tak jak w przypadku innych popularnych mikrokomputerów, magnetofon kasetowy, jednak korzystanie z niego jest tu dużo łatwiejsze. Przede wszystkim do komputera można przyłączyć jednocześnie dwa magnetofony, a ponadto każdy z nich, jeśli tylko ma gniazdo zdalnego sterowania, może być włączany przez komputer. (Doskonale do tego nadaje się produkowany w kraju magnetofon MK 232P). Komputer wyposażony jest ponadto w dwa interface'y szeregowe RS232 o regulowanej szybkości. Pierwszy z nich pozwala na komunikację w dwóch kierunkach (np. można używać NEW BRAIN'a jako inteligentnego terminala — świetnie daje sobie radę np. z Merą 60 czy IBM PC). Drugi, jednokierunkowy, przewidziano do współpracy z drukarką (np. DZM 180 KSR, D 100, STAR SG, SEIKOSHA 250 itp.).

Oprócz tego na tylnej ściance komputera umieszczono 50-cio stykowe złącze dla połączenia zewnętrznych modułów zwiększających możliwości urządzenia.

OPROGRAMOWANIE

Niemal połowa przestrzeni adresowej NEW BRAIN'a jest zajęta przez pamięć stałą. Zawarto w niej system operacyjny, edytor ekranowy, interpreter języka BASIC, pakiet procedur arytmetyki zmiennoprzecinkowej oraz pakiet procedur graficznych. Cechą wyróżniającą system operacyjny NEW BRAIN'a jest elastyczność. Z punktu widzenia programisty wszystkie urządzenia są traktowane w identyczny sposób jako „strumienie" danych. Jednocześnie może być aktywnych nawet 255 strumieni! Zmiana sposobu przypisania strumienia do konkretnego urządzenia (zmiana wartości zmiennej) pozwala bez trudu skierować np. wyniki obliczeń zamiast na ekran, na drukarkę.

Edytor ekranowy NEW BRAIN'a to w zasadzie jądro procesora tekstów. Pozwala on zmienić dowolny znak w dowolnym miejscu ekranu, przesuwać tekst w górę i w dół jeśli rozmiar strumienia ekranu jest większy niż liczba linii, które naraz można wyświetlić, wreszcie rozsuwać znaki w linii, wstawiać ciągi znaków w tekst, dzielić linię tekstu na

KLAN NIETYPOWYCH

Mam komputer CGL-M5 firmy Sord Computer Corporation, produkcji japońskiej. Nigdzie nie mogę zdobyć oprogramowania. Chciałbym nawiązać korespondencję z posiadaczami tego komputera i współpracować w wymianie programów, cartridge'ów itp.

F. Mich
ul. Dekerta 3 m 3
80-262 Gdańsk

fragmenty, które staną się samodzielnymi liniami lub łączyć wiele linii w jedną. Jest bardzo wygodny w użyciu.

Interpreter języka BASIC działa na nieco innych zasadach niż spotykane najczęściej interpretery. Mianowicie realizuje schemat tzw. kompilacji przyrostowej. Polega to na tym, że tekst źródłowy danej linii programu jest przetwarzany tylko raz, przy pierwszym jej wykonaniu. Podczas tego pierwszego wykonania budowany jest w pamięci pewien kod pośredni. Każde kolejne odwołanie do tej linii jest realizowane na podstawie tego właśnie kodu pośredniego. Kolejne wykonania tej samej linii trwają więc krócej niż pierwsze! Dzięki temu właśnie, mimo że dokładność obliczeń jest większa niż na np. ZX Spectrum i wynosi 10 cyfr przy zakresie 10.0E+99, programy na NEW BRAIN'ie działają kilka razy szybciej. (patrz tabelka). Interpreter realizuje podzbiór amerykańskiego standardu języka BASIC ANSI.

Pakiet procedur arytmetycznych zapisany w pamięci ROM może być wykorzystywany nie tylko przez interpreter języka BASIC. W systemie istnieje mechanizm pozwalający wywoływać te procedury z programów napisanych w innych językach np. w assemblerze czy Pascalu. Wszystkie obliczenia zmiennoprzecinkowe prowadzone są na liczbach reprezentowanych przy pomocy 6 bajtów. Dzięki temu można uzyskać, wspomnianą wyżej, dużą dokładność.

Pakiet procedur graficznych wbudowany w pamięć ROM komputera pozwala na wykonywanie wykresów w jednostkach określonych przez użytkownika. Istnieje komenda umożliwiająca wypełnienie dowolnego zamkniętego obszaru. W odróżnieniu od interpretere języka BASIC, procedury graficzne działają dość wolno.

OPROGRAMOWANIE DOSTĘPNE NA RYNKU

Głównym źródłem oprogramowania dla NEW BRAIN'a są obecnie kluby skupiające jego użytkowników. Wśród nich najprężniej działają NEW BRAIN USER GROUP oraz OPEN STREAM w Wielkiej Brytanii. Programy, którymi dysponują, zajmują w sumie kilkanaście dyskietek (a więc kilka megabajtów kodu!). Do najpopularniejszych programów narzędziowych dostępnych dla NEW BRAIN'a należą znany użytkownikom ZX Spectrum debbuger MONS i assembler GENS dla mikroprocesora Z80 — DEVPAC firmy HI-SOFT, kompilator języka Pascal tej samej firmy oraz liczne implementacje języka FORTH. Ponadto, po przyłączeniu sterownika dysków można korzystać z prawie nieograniczonej biblioteki programów systemu CP/M.

CENA

NEW BRAIN zniknął już prawie z rynku brytyjskiego. Nowe egzemplarze (w cenie powyżej 100 funtów) zakupić jest trudno, jednak pełnosprawny komputer z drugiej ręki można zamówić za pośrednictwem prowadzonej przez jednego z twórców NEW Brain'a, Geralda McMullona firmy GFG w Cambridge. Taki zakup nie powinien przekroczyć sumy 60-70 funtów w przypadku modelu AD

●

NEW BRAIN jest nadal interesującą propozycją dla tych, którzy chcą posługiwać się komputerem domowym jako narzędziem do rozwiązywania problemów. Doskonała grafika, liczne wbudowane interface'y i duża dokładność obliczeń czynią z niego niemal idealne narzędzie wspomagające prace inżynierskie, laboratoryjne, opracowywanie wyników eksperymentów itp.

Jan Hałemejko

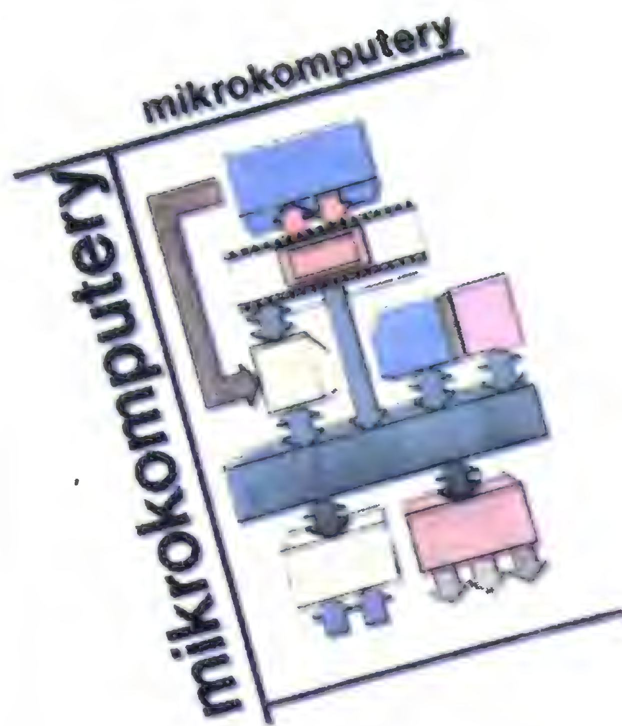
UŻYTKOWNICY NEW BRAIN'a w Polsce

W Warszawie od ponad dwóch lat działa nieformalna grupa użytkowników NEW BRAIN'a. Liczy ona, jak dotąd czterech członków. Utrzymujemy stały kontakt z klubami entuzjastów tego komputera w Wielkiej Brytanii.

Wszystkich zainteresowanych „poważnymi" zastosowaniami mikrokomputerów zapraszamy do współpracy!

PRZECZYTALIŚMY DLA WAS

„MIKROKOMPUTERY"



„MIKROKOMPUTERY" (tytuł oryginału: **„Microcomputers"**) pod redakcją **A.J. Dirksena**, wydane w marcu br. nakładem **Wydawnictw Komunikacji i Łączności** — to praca zbiorowa autorów holenderskich. Ukazała się ona w 1978 roku w języku holenderskim, ale duża popularność sprawiła, iż szybko ukazały się jej przekłady na język angielski i rosyjski.

W dwudziestu jeden rozdziałach autorzy — w sposób przystępny i poglądowy — przedstawili pokrótce wszystkie zagadnienia z dziedziny mikrokomputerów. Wytłumaczyli więc na wstępie co to jest komputer i czym się od niego różni mikrokomputer, jak komputer liczy, jakie posiada układy logiczne, co to jest pamięć operacyjna i na czym polega jej organizacja, jak zbudowane są moduły i jak odbywa się adresowanie. Następnie autorzy wprowadzają nas w tajniki programowania wyjaśniając, na czym polega architektura procesora i mikrokomputera, jak należy pojmować opis rozkazów, na czym polega syntaktyka i podprogramy, jakie są tryby adresowania i sieci działań, wreszcie — jak sformułować i rozwiązać jakieś zadania, zorganizować komunikację mikrokomputera z urządzeniami zewnętrznymi itp.

W trakcie lektury książki zaznajamiamy się bliżej z różnymi pojęciami, np.: komparator, arytmometr, edytor tekstowy, asembler, debugger, symulator, kompilator i z wieloma innymi, nie mającymi ustalonych i powszechnie zaakceptowanych odpowiedników w języku polskim. Dlatego też, aby ułatwić czytelnikowi zaznajomienie się ze stosowaną terminologią zamieszczono na końcu książki skorowidz używanych w tekście pojęć w wersji dwujęzycznej. Oprócz tego dołączony także został spis literatury uzupełniającej, wydanej do tej pory w języku polskim oraz — do ilustracji omawianych zagadnień — opis listy rozkazów mikroporcesora 8080 i szczegółowy opis ich wykonywania.

Pod koniec każdego rozdziału zamieszczone zostało podsumowanie omawianego zagadnienia, uzupełnione o pytania sprawdzające i odpowiedzi.

Książka jest swoistą kopalnią wiadomości o komputerach, ich architekturze i działaniu. Szkoda tylko, że jej nakład jest na prawdę mikro: 9800 egzemplarzy!

(JZ)

„Mikrokomputery" pod red. A.J. Dirksena, z jęz. ang. tłumaczyli Maria i Marek Gondzio, Wyd. Komun. i Łączn., Warszawa 1985, wyd. I, stron 328, nakład 9800 egz.

AMstudio

Pewnego listopadowego popołudnia 1985 roku spotkało się trzech świeżo upieczonych właścicieli AMSTRADA. Nie mieliśmy zbyt dużo oprogramowania, dwie pozycje z literatury i prawie żadnych kontaktów. (Iluż ludzi musi być w podobnej sytuacji). Postanowiliśmy założyć klub.

Zaczęliśmy od zalegalizowania i określenia zakresu działalności klubu. Jesteśmy studentami, w związku z czym nawiązaliśmy współpracę z Zespołem Mikrokomputerowym działającym przy poznańskim Student Serwisie. Na początku grudnia byliśmy już prawie usankcjonowani. Dowiedzieliśmy się o organizowanej w Warszawie wystawie Mikroexpo 85. Postanowiliśmy pojechać na wystawę, ogłosić zawiązanie się pierwszego w Polsce klubu AMSTRADA/Schneidera — AMstudio. Można powiedzieć, że od tego właśnie momentu zaczęła się prawdziwa działalność klubowa tj. m.in.: wymiana oprogramowania, literatury, doświadczeń, pokazy komputerowe, artykuły, wywiady. Taki był początek.

Obecnie dysponujemy 25-ma pozycjami z literatury w tym także kilkoma przekładami na język polski. Do tej pory udało nam się zgromadzić ok. 500 programów, w tym liczne języki programowania: PASCAL, TURBO-PASCAL, FORTH, FORTRAN, HISOFT C, LOGO oraz wiele odmian assemblera — Zen assembler, Devpac assembler, Honey assembler, Arnor assembler, ZAAP (Z80 assembler), CPC assembler.

Prenumerujemy następujące pisma: Amstrad Computer User, Amstrad Magazine, Amstrad Action, Computing with Amstrad, Micro Strad, Schneider Activ. Utrzymujemy kontakty z licznymi klubami, osobami, a także firmami zagranicznymi głównie w Wielkiej Brytanii, Francji i RFN.

Wydajemy pismo klubowe, w którym czytelnicy mogą znaleźć m.in. następujące działy: forum użytkowników, nowości hard/soft, ceny sprzętu na rynkach zagranicznych i krajowych oraz wydruki programów. Poza tym własnym, polskim opracowaniem Basica na Amstrada pt. „Basic od podstaw", które składa się z książki i kasety z przykładowymi programami i testami sprawdzającymi wiedzę programisty.

AMstudio to nazwa określająca nie tylko klub, lecz także małą firmę software'ową. Z członków klubu utworzyliśmy zespół zajmujący się wyłącznie pisaniem oprogramowania edukacyjnego. Mamy nadzieję, że w niedalekiej przyszłości zalety Amstrada zostaną docenione przez szkolnictwo i praca nasza nie pójdzie na marne. W związku z tym, że klub nasz posiada pozwolenie na sprzedaż, postanowiliśmy rozpowszechnić własne programy, napisane przez członków klubu.

Na początek proponujemy sześć zestawów:

1. Zestaw programów narzędziowych (m.in. generator symboli, program kopiujący i przyśpieszający proces ładowania, monitor-disassembler, udoskonalenie funkcji TRON, hardcopy z odcieniami szarości)
2. Zestaw programów użytkowych (m.in. baza danych, książka telefoniczna, program kalkulacyjny, zegar, wykresy)
3. Zestaw programów domowych (m.in. dieta, jak się odchudzać, budżet domowy, biorytmy, test dla palaczy, jakim jesteś kierowcą)
4. Zestaw programów edukacyjnych dla dzieci
5. Zestaw gier
6. Basic — demonstracja + testy + instrukcja.

Do AMstudio może zapisać się każdy użytkownik komputera AMSTRAD/Schneider. Żeby zostać członkiem naszego klubu należy przesłać pod następujący adres: **PIOTR DERĘGOWSKI ul. Żwirki 10a, 60-409 Poznań AMstudio** informacje dotyczące typu swojego komputera, dane personalne, dokładny adres oraz zgłoszenie wymiany lub kupna (np. pisma klubowego).

W odpowiedzi prześlemy kartę klubową wraz z numerem ewidencyjnym. Każdy członek naszego klubu ma prawo dostać 2-3 programy rozrywkowe na początek.

Listę programów, którymi dysponuje klub przesyłamy na życzenie. W przypadku korespondencji prosimy dołączyć znaczek zwrotny. Z góry przepraszamy za ewentualną zwłokę w odpowiedzi na listy, liczba AMSTRAD'ów w Polsce jest szacowana na ok. 14 tys. a nas jest tylko kilku.

Piotr Deręgowski



KORESPONDENCJA Z WIEDNIA

Nad pięknym, modrym Dunajem

W ostatnim czasie w Polsce rozpoczął się „boom" komputerowy i trwa szeroka dyskusja zarówno o komputeryzacji szkół, jak i o wprowadzaniu komputerów do życia codziennego. W wielu krajach zachodnich, m.in. w Austrii już przed kilkoma laty zorientowano się, iż komputery staną się niezbędne w przyszłości i rozpoczęto, z o wiele mniejszym entuzjazmem niż w Polsce, ich popularyzację. Nastąpiła ona jednak tu o wiele szybciej i w zasadzie samoczynnie.

Kiedy komputery, poprzedzone kampanią reklamową, zaczęły ukazywać się w sklepach, wzrosło zainteresowanie nimi, przede wszystkim wśród młodzieży. Nie istniała i nie istnieje tutaj największa bariera hamująca popularyzację komputerów w Polsce, mianowicie cena. Np. komputer klasy **Commodore 64** kosztował zaraz po ukazaniu się w sprzedaży od 7 do 9 tysięcy szylingów, podczas gdy pensja przeciętnego obywatela wynosiła ok. 10 tys. szyl. Oczywiście w ciągu ostatnich lat wszystkie komputery staniały kilkakrotnie i są w cenach bardzo przystępne. Komputer tej klasy kosztuje obecnie od 4 do 5 tys. szyl. Dosyć duża część młodzieży, przede wszystkim męskiej, posiada je w domach, a w szkołach organizowane są ponadobowiązkowe, bezpłatne zajęcia komputerowe. Od obecnego roku szkolnego podstawy informatyki są także obowiązującym przedmiotem w klasie odpowiadającej naszej 1 kl. lic.

Komputery szkolne to komputery osobiste (Personel Computers) — w mojej szkole mamy komputery firmy **Honeywell Bull** model **Micral 30** kompatybilne z **IBM-PC.** Każdy wyposażony jest w kolorowy monitor, drukarkę oraz podwójną stację dysków. Posiadamy w szkole 7 takich stanowisk.

W telewizji — podobnie jak w Polsce — można oglądać audycje dotyczące zagadnień informatycznych oraz pracy na komputerze. Na rynku jest do nabycia szeroka paleta czasopism komputerowych — większość z nich jest w języku niemieckim, ale importowane są też wydawnictwa angielskojęzyczne, jak np. **„Byte"** czy **„Your Computer"** — znane też w Polsce. Tematyka czasopism obejmuje zarówno proste zagadnienia dla początkujących, jak i problemy interesujące zaawansowanych. Są one pomocą dla młodego użytkownika **Sinclair'a i ZX 81**, a równocześnie i dla managera obsługującego **IBM**.

O ile w Polsce przeważają komputery domowe, używane przede wszystkim do gier i prostych programów, o tyle tu rozwinięty jest bardzo rynek komputerów profesjonalnych używanych powszechnie w przedsiębiorstwach. Większość instytucji, zarówno prywatnych jak i państwowych, a także wiele sieci sklepów i marketów jest już skomputeryzowanych. Komputery domowe stanowią dlatego niewielką część ogółu sprzedaży.

Austriacy nie produkują prawie wcale własnych komputerów ani oprogramowania do nich. Wszystko to importowane jest z USA, RFN, W. Brytanii oraz Japonii. Programy nabyć można w specjalnych sklepach komputerowych lub w domach towarowych i są one przeważnie w języku angielskim, co nie stanowi jednak zbyt wielkiej trudności, gdyż jest on tu powszechnie znany (obowiązkowe nauczanie od kl. 5).

Gry komputerowe są artykułem sprzedającym się bardzo źle. Powodem tego jest ich cena (dobra, nowa gra na kasecie lub dyskietce kosztuje w przeliczeniu około 30 dol.) oraz powszechnie uprawiane piractwo. Chociaż w ostatnim czasie wprowadzono bardzo ostre kary za ten proceder, istnieje rozwinięte „podziemie komputerowe" i w kilka tygodni po ukazaniu się programu można go uzyskać za cenę wielokrotnie mniejszą. Ja sam nie znam żadnego użytkownika komputera, który by kupował oryginalne programy.

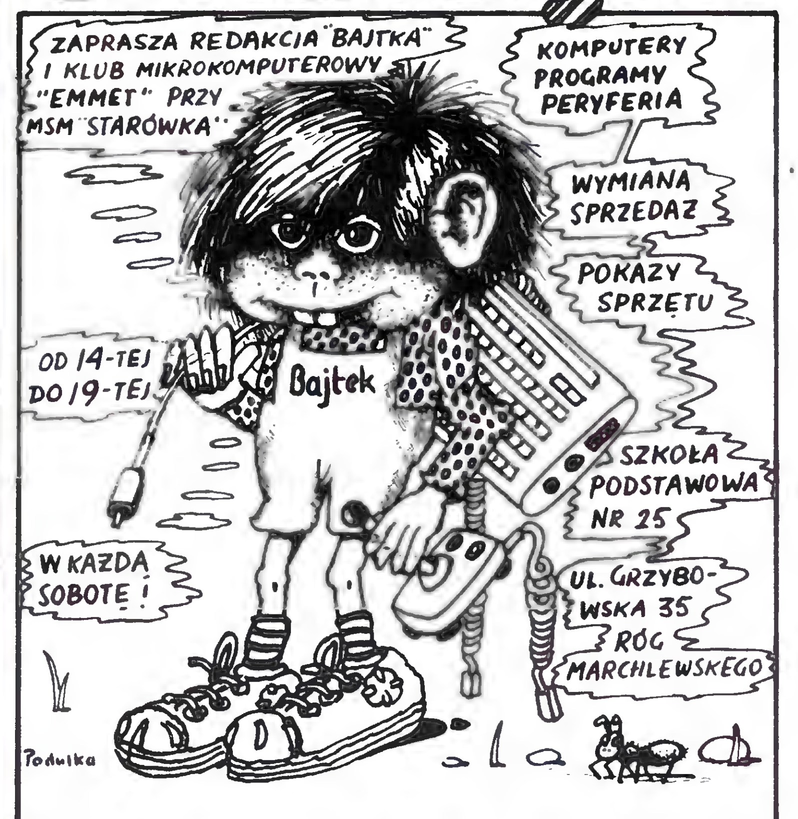
W Austrii istnieje pokaźna liczba klubów komputerowych, w których wymieniać można doświadczenia i programy (oczywiście jest to także niezgodne z prawem). Przynależność do danego klubu związana jest z płaceniem składki członkowskiej, co jednak jest sprawą opłacalną, ponieważ zapewnia dostęp do wielu typów komputerów, czasopism, programów oraz możliwość ich kopiowania.

Najbogatsze oprogramowanie i najwięcej literatury fachowej można znaleźć do komputera roku 1983 i 1984, a mianowicie do **Commodore 64**. Jest on z pewnością najpopularniejszym komputerem domowym, zarówno w Austrii, jak i w RFN. Istnieje do niego także bardzo dużo przystawek, umożliwiających np. sterowanie kolejką elektryczną, połączenie z kamerą video, prowadzenie obserwacji meteorologicznych, lub nawet kontrolowanie procesu warzenia piwa(!).

Drugim jeżeli chodzi o popularność komputerem jest **Atari 800 XL** oraz nowe modele **Atari**. **Sinclair Spectrum** — w Polsce tak popularny — tu obecnie wychodzi ze sprzedaży i kupić go można już nawet za 120 dolarów (wersja 48K).

Wojtek Stankiewicz (lat 16)

		BAJTKOWA GIEŁDA (tys. zł)	KOMIS (tys. zł)	BOMIS (tys. zł)	BALTONA REWEX (Dol. USA)	RFN (średnie) (DM)	WLK. BRYT. (średnie) (£)
SINCLAIR	ZX 81	25–30	30	—	—	90	—
	ZX Spectrum 48 kB	80–90	110	130	125	260	65–70
	ZX Spectrum Plus	110–115	150	180	—	320	80–90
	ZX Spectrum 128 kB	195	—	—	—	598	160
	Drukarka SEIKOSHA GP50S	90	100	150	—	260	60–65
	Układ specjalizowany ULA	12	—	—	—	—	7
	Z-80 A	—	—	4	—	4	1.4
COMMODORE	C-64	140–150	150–160	240	—	450–520	90–110
	C-128	300–400	450	450–500	—	700	210–230
	C-128 D	—	—	—	—	1500–1500	390–410
	Magnetofon 1531	20–30	35	—	—	80–100	—
	Stacja dyskietek 1541	150–160	200	350	—	520	110–150
	Stacja dyskietek 1570	200	240	700	—	490–540	160–170
	Drukarka MPS 801	120–130	150	280	—	300	—
	Drukarka MPS 803	—	250	360	—	330	140–150
	Dyskietki 5 1/4 (średnia jak.)	0.9–2	1.5	2–3.5	—	19–60	1–2
ATARI	800 XL	70–80	90	—	115	189	70–80
	130 XE	150	—	380	199	378	150
	Stacja dyskietek 1050	130–140	200	—	185	478	130
	Magnetofon	25–30	35	—	48	75	20
	Drukarka 1029	140	—	—	199	—	—
AMSTRAD	464 z mon. monochromat.	220–245	300	500	—	680–720	160–180
	464 z monitorem kolor.	310	—	600	—	1100–1200	260–268
	664 z mon. monochromat.	300–350	—	600–700	—	1000–1300	—
	6128 z mon. monochromat.	385–400	700	850	—	1450–1500	250–260
	6128 z mon. kolor.	500–550	—	1.1 mln.	—	1900–2000	340–350
	PCW 8256	1 mln.	1.5 mln.	1.2 mln.	—	2000–2100	380–400
	Dyskietki 3"	5–5.5	6–7	8	4	10	3.5–4



EMMET

Nasz klub powstał przypadkiem. Po prostu wraz z grupką przyjaciół oraz mym ojcem i bratem postanowiliśmy kiedyś założyć klub mikrokomputerowy. Pomysł rzucony był żartem, ale padł na podatny grunt, bowiem wszyscy byliśmy i jesteśmy zafascynowani mikrokomputeryzacją. Traktujemy ją jako klucz do lepszego świata. Jest to może podejście nieco filozoficzne, ale chcielibyśmy jakoś pomóc tym dziesiątkom młodych i piekielnie zdolnych młodych ludzi. Przecież można dać im sprzęt, który po przejściu etapu oczarowania, stanie się normalnym, zwyczajnym narzędziem pracy. Jest to bardzo ważne, bo przecież dzisiejsze dzieci jutro będą zarządzały przemysłem polskim, będą musiały stawić czoła światowym potęgom technicznym i technologicznym, a wtedy już nie da się kierować przedsiębiorstwem przy pomocy stu księgowych, stu pięćdziesięciu znerwicowanych zaopatrzeniowców i ton papierów przewalających się przez biurka. Trzeba dać tym ludziom jakąś szansę. Wielu z nich wydaje się, że nauka to marnowanie czasu, bo po byle zawodówce zarobi jak docent, a kiedy weźmie fuchę to przegoni dwóch profesorów. Jest to sposób myślenia cokolwiek niepoważny, ale to trzeba pokazać.

Na początku kwietnia rozpoczęliśmy organizowanie własnej giełdy mikrokomputerowej. W maju patronat nad naszą giełdą przejęła redakcja BAJTKA.

Skąd wziął się pomysł. Po pierwsze: klub musi mieć możliwość zarobienia pieniędzy na zakup choć minimalnego zestawu sprzętu, a po drugie — giełda to znakomite miejsce do wymiany oprogramowania, zbierania informacji o nowinkach technicznych i wreszcie świetna okazja do obejrzenia z bliska najnowszych modeli mikrokomputerów. Tak, tak, to nie pomyłka, na jednej z pierwszych giełd gościł — wtedy nowość absolutna, dwa tygodnie na rynku angielskim — świeżutki brytyjski Spectrum 128.

Na naszej giełdzie — a mamy ambicję by była to największa tego typu impreza w Warszawie — na trzech piętrach szkoły możemy swobodnie zapewnić 60 stanowisk software'owych i drugie tyle hardware'owych. Wprowadziliśmy możliwość rezerwacji stolików — nawet na miesiąc z góry, giełda ma bufet, gdzie obecny brak ciepłych posiłków rekompensują słodkości, a kawę czy zimne napoje może dostojnie popijać nawet 50 spragnionych.

Na każdej giełdzie jest osobne stoisko BAJTKA. W maju prezentowaliśmy sprzęt wypożyczony przez Redakcję i okazało się, że młodsi adepci klubu znakomicie dawali sobie radę z obsługą sprzętu klasy „COMMODORE 128" czy „AMSTRAD 6128". Dlatego też korzystając z okazji chciałbym zaprosić wszystkie firmy chcące zaprezentować swój sprzęt — na naszą giełdę. Możemy zapewnić stoisko oraz ewentualnie obsługę.

Na stoisku BAJTKA można oczywiście również kupić także wcześniejsze numery tego miesięcznika (w nowej szacie graficznej; małe są już wyczerpane). Natomiast przy stoisku organizatorów można skorzystać ze swego rodzaju biura pośrednictwa (nieodpłatnie) oraz zasięgnąć informacji o niemal wszystkim co dotyczy mikrokomputerów, peryferiów i oprogramowania tak w Polsce jak i na świecie. Zapraszamy więc w każdą sobotę, w godzinach 14-19 na ulicę Grzybowską 35 (róg Marchlewskiego).

Jan Popończyk
Prezes klubu mikrokomputerowego „emmet"

Adres do korespondencji:
00-958 Warszawa 66
skrytka pocztowa nr 2

- SPRZĘT MINIKOMPUTEROWY
- URZĄDZENIA PERYFERYJNE
- OSPRZĘT I OPROGRAMOWANIE
- SPRZĘT MAGNETOWIDOWY

Skup i sprzedaż w/w artykułów prowadzi nowouruchomiony dział w sklepie 163 WPHW, Oddział Dąbrowa Górnicza
Sklep prowadzi sprzedaż pozarynkową

ZAPRASZAMY

do sklepu nr 163

„ELEKTRON"

Dąbrowa Górnicza,
ul. Sobieskiego 17
w godz. 10^{00} — 18^{00}, tel. 62-23-71

Oferujemy niskie ceny detaliczne np.:

1. ZX SPECTRUM	- 122.000,-
2. ZX SPECTRUM PLUS	- 170.000,-
3. ZX SINCLAIR QL	- 380.000,-
4. DRUKARNIA GP-50	- 145.000,-
5. Commodore C 64 + mag	- 250.000,-
6. Commodore C 128	- 400.000,-
7. Atari 800 XL + mag	- 170.000,-
8. Atari 130 XE	- 210.000,-
9. Schneider CPC 464 z ziel. mon.	- 370.000,-
10. Schneider CPC 6128 z ziel. mon.	- 880.000,-
11. Sharp MZ 731 kpl	- 270.000,-

oraz na inne urządzenia peryferyjne i oprogramowanie.

G-60

OFERTA

dla Przedsiębiorstw, Instytutów, Placówek Naukowo-Badawczych, Szkół

„BOMIS"

PRZEDSIĘBIORSTWO OBROTU MASZYNAMI I SUROWCAMI
60-693 POZNAŃ, UL. SKOCZOWSKA 19,
TELEX 0413606

PROWADZI CIĄGŁĄ SPRZEDAŻ:

1. Mikrokomputer ZX Spectrum 48 kb Standard	176.000,-
2. Mikrokomputer ZX Spectrum Plus-48 kB	242.000,-
3. Mikrokomputer „Schneider" CPC 664 z monitorem monochrom. kolor.	955.000,- 1.295.000,-
4. Mikrokomputer j.w. CPC 6128 z monitorem monochrom. kolor.	1.091.000,- 1.420.000,-
5. Drukarka GP-50S Seikosha	177.000,-
6. Drukarka GP-25-ox Seikosha	333.000,-

UWAGA!!! UWAGA!!!

TYLKO U NAS!

Na pozycje 1-4 Przedsiębiorstwo udziela 12-miesięcznej gwarancji na układy elektroniczne, natomiast na układ mechaniczny mikrokomputera „Schneider" 6-miesięcznej.
Zamówienia realizowane będą wg kolejności ich wpływu.
Informacji udziela Dział Sprzedaży „Bomisu", tel. 77-24-13.
S-62

SZEROKI ZAKRES USŁUG W ZAKRESIE KOMPUTERÓW PROFESJONALNYCH (IBM-PC/XT, SCHNEIDER):

— SPRZEDAŻ SPRZĘTU
— SERWIS HARDWARE'OWY I SOFTWARE'OWY
— OPROGRAMOWANIE
— CZĘŚCI ZAMIENNE
— PORADY

ŚWIADCZY FIRMA

PROMOTOR

PROSIMY TELEFONOWAĆ NR 13-65-01 W GODZ. 10^{00} — 16^{00}.

K-104

Instytut Maszyn Matematycznych zatrudni natychmiast:

inżynierów elektroników

Zgłoszenia przyjmuje i bliższych informacji udziela Dział d/s Pracowniczych Instytutu, Warszawa, ul. Krzywickiego 34, tel. 21-50-97 lub 21-84-41 w. 544.
K-102

SPÓŁDZIELNIA RZEMIEŚLNICZA

„PRODUCENT"

18-400 Łomża, ul. Nowogrodzka 200, tel. 60-62.

Oferuje do sprzedaży:

1. Interface Sinclair do Zx Spectrum.
2. Interface Kempston do Zx Spectrum.
3. Joystick.

Ceny konkurencyjne, krótkie terminy.
ROCZNA GWARANCJA!!!

K-103

Firma

MUEL

oferuje do sprzedaży:

1) Interface do ZX-Spectrum umożliwiający współpracę z czterema napędami dysków elastycznych, dowolną drukarką graficzną, monitorem ekranowym, rozszerzający BASIC oraz system operacyjny ZX-Spectrum. Nie zajmuje pamięci RAM!
2) Sterowany „ikonami" programator EPROM 2716 + 27256 do ZX-Spectrum
3) Przeróbkę drukarki DZM 180 na drukarkę graficzną.

- Informacja tel. 33-40-91
- Korespondencja: MUEL ul. Cząstkowska 30 01-678 Warszawa

K-109

Atari. Interface do zwykłego magnetofonu zastąpi magnetofon fabryczny. Informacje: Warszawa 34-16-06.
D-83

PROGRAMY KOMPUTEROWE POCZTA: dla ATARI, AMSTRADA, COMMODORA, i SPECTRUM wysyła Agencja Mikrokomputerowa Sosnowiec P-157, tel. 699-649.
K-76

SPECTRUM — programy wysyła: SPECTRA, 21-426 Wola Mysłowska.
D-86

JAK SIĘ REKLAMOWAĆ W BAJTKU?

Reklamy przyjmuje Młodzieżowa Agencja Wydawnicza (Redakcja Wydawnictw Poradniczych i Reklamy), 04-028 Warszawa, Al. Stanów Zjednoczonych 53, pokój 313. tel. 10-56-82

Cena ogłoszeń: 200 zł za 1 cm plus dodatki za kolor



Drogi Bajtku!

Czy interface to „przedłużacz z rozwidlaczem", do którego podłącza się wszystkie urządzenia peryferyjne?

Co oznacza „1 napęd lub 2 napędy stacji dysków"?

Co oznacza stwierdzenie: „64 kB RAM, 24 kB ROM, w tym 8 kB Basic i 16 — system operacyjny"?

Co oznacza termin: „rozdzielczość 320 x 192"?

Maciej Górski
ul. Smętka 20a/23
Giżycko

Termin „interface" jest używany do określenia połączenia oraz powiązanych z nim układów elektronicznych, zapewniającego wymianę informacji pomiędzy jednostką centralną (procesorem) komputera oraz jego urządzeniem peryferyjnym (lub urządzeniami peryferyjnymi). Tę definicję przytaczam za książką A. Chandora „A Dictionary Of Computers", Penguin Books 1972. Z reguły różne urządzenia peryferyjne wymagają różnych interface'ów. Należy też pamiętać, że podłączenie każdego urządzenia peryferyjnego wymaga użycia interface'u; klawiatura, monitor czy magnetofon kasetowy nie są tu wyjątkami — w ich przypadku jednak, potrzebne układy umieszczone są wewnątrz obudowy komputera.

Przyjęło się używać pojęcia „1 lub 2 napędy dysków" — trzeba przyznać dość niefortunnego — dla określenia ilości dysków, z jaką można jednocześnie pracować. Np. w przypadku IBM PC termin „dwa napędy dysków" oznacza, że komputer wyposażony jest w stację dysków, do której można jednocześnie włożyć dwie dyskietki.

Przytoczony cytat oznacza dokładnie: „64 kB (1 kilobajt = 1 024 bajty) pamięci o swobodnym dostępie, 24 kB pamięci stałej, w której 8 kB zajmuje interpretator języka Basic, a 16 kB — system operacyjny".

Dana rozdzielczość ekranu określa ilość punktów świetlnych, na jaką ekran jest podzielony. W podanym przykładzie rozdzielczość 320 x 192 oznacza 320 punktów świetlnych wzdłuż osi poziomej ekranu i 192 takie punkty — wzdłuż osi pionowej. Oczywiście im większa rozdzielczość, tym lepsza jakość obrazu generowanego przez komputer.

Od dwóch tygodni próbuję swoich sił, jako programista w LOGO. Zgadzam się, że to doskonała zabawa. Przy próbie rysowania wykresów funkcji natrafiłem jednak na pewien problem, z którym jak dotąd nie mogę sobie poradzić. Otóż przy próbie użycia instrukcji SETPOS w programie występuje błąd:

„SETPOS doesn't like [: x : y] as input"

z drugiej strony jednak instrukcja niekiedy działa, np.

SETPOS [50 50]

Czy mógłbym prosić o wyjaśnienie, dlaczego tak się dzieje?

M.B.
(nazwisko i adres do wiadomości redakcji)

Prawdopodobnie próbował Pan użycia SETPOS w postaci:

SETPOS [:x :y]

W tym przypadku jednak zapis [:x :y] oznacza listę złożoną z dwóch słów: ":x i ":y. Sygnalizuje to przytoczony komunikat LOGO. Podobnie niepowodzeniem zakończyłaby się np. próba użycia

SETPOS [XCOR Ø]

— można spróbować! Aby uzyskać pożądany skutek, należy użyć

SETPOS LIST :x :y

lub

SETPOS SE :x :y

w tym wypadku bowiem listy utworzone przez LIST i SE są identyczne.

Podczas wykonywania rysunku w LOGO (na CPC 6128) niekiedy pojawiają się dwa żółwie — jeden z nich jest nieruchomy, drugi zaś wykonuje wszystkie wprowadzane z klawiatury polecenia. O czym to świadczy — czy tym drugim żółwiem można również w jakiś sposób kierować?

Krzysztof Wyrzykowski
(adres do wiad. redakcji)

Jest to niestety, błąd interpretera. Wersja Dr LOGO rozpowszechniana przez firmę Digital Research operuje jednym żółwiem; rozdwajanie się jego wizerunku spowodowane może być np. niewłaściwą realizacją procedur ht (hideturtle) i st (showturtle). Nie jest to jedyny rzucający się w oczy błąd Dr LOGO 2 i 3.

Niektóre wersje LOGO faktycznie umożliwiają posługiwanie się kilkoma żółwiami, sterowanymi osobno — taką możliwość mają np. użytkownicy LOGO na Atari 800XL.

W jakim stopniu Amstrad CPC 6128 jest kompatybilny z CPC 464?

Leszek Kordyl
Wejherowska 26/12
81-042 Gdynia

Wszystkie programy używane na maszynie CPC 464 mogą być bez przeszkód uruchomione na CPC 6128. Komputer ten jest również wyposażony przez producenta w dyskietkę z systemem CP/M 2.2 (obok jego najnowszej wersji 3.0), zatem istnieje możliwość wykorzystania całego programowania używanego na CPC 464.

Mój komputer (Philips VG 8010) nie wybiera liczb losowych. Czy tę wadę da się w jakiś sposób wyeliminować?

Artur Chodacki
Os. XX-lecia 13/22
34-300 Żywiec

Najprościej prawdopodobnie zaprogramować generator liczb pseudolosowych samemu. Istnieje bardzo wiele metod realizacji ciągu liczbowego, który miałby cechy zbliżone do ciągu liczb losowych (o rozkładzie jednostajnym). Jedną z najprostszych, i raczej wystarczających do mniej skomplikowanych zastosowań, jest ciąg liczb naturalnych x^n postaci:

$$x_n = (c_1x_1 + c_2x_2 + c_3x_3 + ... + c_{n-1}x_{n-1}) \bmod n$$

gdzie liczby $c_1...c_{n-1}$ są pewnymi stałymi współczynnikami. Przykładem takiego ciągu może być choćby nieco zmodyfikowany ciąg Fibonacciego:

$$x_1 = p_1 \qquad x_2 = p_2$$
$$x_n = (c_{n-1}x_{n-1} + c_{n-2}x_{n-2}) \bmod n$$

gdzie p_1, p_2 — pierwsze dwa wyrazy ciągu,
c_1, c_2 — stałe współczynniki
działanie a mod b oznacza resztę z dzielenia a przez b

Tak ustalony ciąg jest ciągiem okresowym; należy tak dobrać jego współczynniki, aby okres był możliwie jak najdłuższy, ze względów zrozumiałych (np. przy okresie 1 ciąg jest stały). Otrzymane x_n możemy wówczas traktować jako liczbę naturalną wybraną losowo spośród liczb 0..n-1. Wartości kolejnych „losowań" zależą od wyboru stałych p_1 i p_2 — zatem należy zadbać o nadanie im odpowiednich wartości przed uruchomieniem programu.

Więcej szczegółów na ten temat, dotyczących np. właściwego doboru współczynników i innych metod generacji ciągów liczb pseudolosowych znaleźć można w książce J. Zielińskiego „Generatory liczb losowych".

Mam możliwość używania komputera CPC 664. Moim zamierzeniem było programowanie w Turbo-Pascalu. Wszystko szło dobrze, do momentu... próby wyjścia z edytora. Po napisaniu tekstu programu nie mogę przerwać pracy w edytorze i przejść do kompilacji. Próbowałem chyba wszystkich możliwych kombinacji klawiszy, ale bezskutecznie. Nie mam oryginalnej instrukcji obsługi tego programu, zaś funkcje edytora nie są opisywane na ekranie (jak to ma miejsce np. w programie Tasword). O ile to możliwe, prosiłbym o rozwiązanie tego problemu — umożliwiłoby to mi pracę na komputerze.

Z.G.
(nazwisko i adres do wiad. redakcji)

Turbo-Pascal wyposażony jest w wygodny edytor, jednak jego obsługa bez znajomości podstawowych funkcji może być trudna. Wyjście z edytora powoduje wciśnięcie klawiszy CTRL-K CTRL-D (w tej kolejności). Oto niektóre inne jego funkcje:

CTRL-V Przełączenie trybu wstawiania/zakrywania znaków
CTRL-Y Usunięcie linii, w której znajduje się kursor
CTRL-R Poprzednia strona tekstu
CTRL-C Następna strona tekstu
CTRL-Q CTRL-R Początek tekstu
CTRL-Q CTRL-C Koniec tekstu
CTRL-K CTRL-B Zaznaczenie początku bloku w miejscu kursora
CTRL-K CTRL-K Zaznaczenie końca bloku w miejscu kursora
CTRL-K CTRL-W Zapisanie bloku na dysk
CTRL-K CTRL-R Wczytanie bloku z dysku
CTRL-K CTRL-C Kopiowanie bloku
CTRL-K CTRL-Y Usunięcie bloku

Marcin

NIE BÓJ SIĘ MNIE

SUPEŁEK NA TRĄBIE

Czytaliście o kłopotach Słonia Trąbalskiego? Biedak, spotykały go same przykrości z powodu słabej pamięci. Nie potrafił zapamiętać nawet imion własnych dzieci...

Jak to się dzieje, że pamiętasz jak ma na imię twój kolega, ile będziesz miał dzisiaj lekcji, na którą godzinę umówiłeś się z koleżanką? Zapisywanie i przechowywanie informacji w twoim mózgu jest procesem niezwykle skomplikowanym, w moim przypadku jest to o wiele prostsze, choć zasada pozostaje ta sama.

W wyniku swego roztargnienia pan Trąbalski, zamiast do lekarza, trafił do kowala, który dał mu dobrą radę: zawiązać sobie supełek na trąbie. Rada była rzeczywiście dobra, ale niestety okazała się nieskuteczna — po prostu Trąbalski zapomniał, w jakim celu zawiązał sobie tenże supełek.

— Gdzie więc tkwił błąd?

Otóż pan Trąbalski powinien podzielić swoją trąbę na odcinki, z których każdy oznaczałby jakąś rzecz do zapamiętania. I tak na przykład pierwszy odcinek — począwszy od końca trąby — to zakupy, drugi — imieniny żony, trzeci — wizyta gości. Teraz wystarczy zawiązać supełek na odpowiednim miejscu i sprawa załatwiona. No i jeszcze jedno, panie Trąbalski. Warto zapisać sobie co oznacza każdy odcinek trąby!



Spójrzmy teraz na nasze słonie. Pierwszy jest wyraźnie bardzo zadowolony, bo nie ma nic do roboty, drugi ma nieco smutniejszą minę — supełek na środku trąby to zakupy, trzeci zaś wygląda bardzo żałośnie, ponieważ musi się wybrać do dżungli po banany dla gości, którzy przybędą na imieniny żony. Taki oto pożytek można mieć ze zwykłej trąby. Przypuszczam zresztą, że większość posiadaczy trąb nie zdaje sobie z tego sprawy!

Zastanawiacie się pewnie, czemu — zamiast mówić o sprawach poważnych — opowiadam nieprawdopodobne historie o roztargnionych słoniach. Zapewniam was, że mówiłem zupełnie poważnie, a moja rada dla Trąbalskiego to nie żart. Ja — Wasz Komputer, dokładnie w ten sam sposób zapamiętuję wszystkie informacje. Oczywiście nie mam trąby i nie wiążę supełków, ale wszystko zapisuję w pamięci w ciągu jedynek i zer, co właśnie odpowiada zawiązaniu lub nie supełka na trąbie. Taki supełek stanowi podstawową jednostkę informacji, tzw. bit. Znacznie łatwiej jest mi jednak posługiwać się nieco większą jednostką, jaką jest bajt (u mnie składa się na niego osiem bitów). Bajt, czyli trąba mieszcząca osiem supełków, pozwala zapamiętać np. liczbę z przedziału od 0 do 255, lub dowolny znak dostępny z klawiatury.

Pojemność mojej pamięci podaje się zazwyczaj w jeszcze większych jednostkach — kilobajtach. Jeden kilobajt oznacza 2^{10}, czyli 1024 bitów. Największą stosowaną jednostką jest megabajt, tj. — 1024 × 1024 bajty.

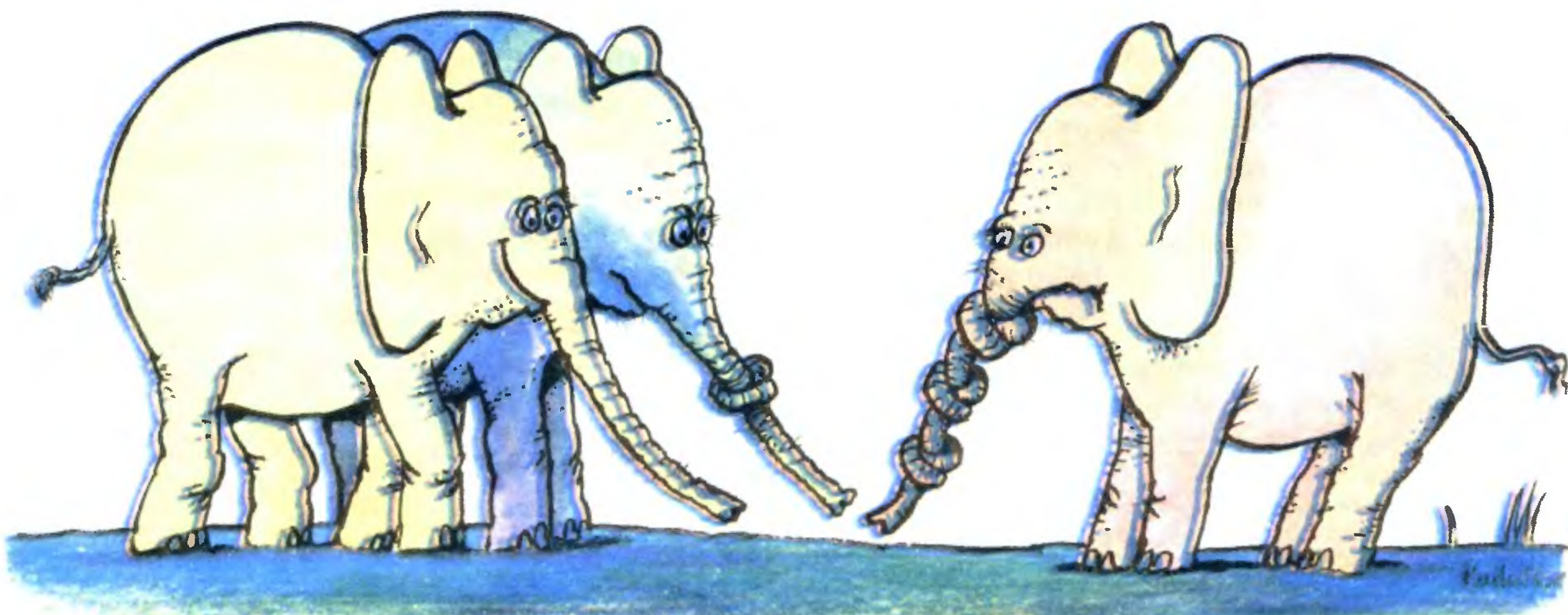
Każdy komputer natychmiast po włączeniu posiada już pewną ilość informacji i umiejętności. Dzieje się to za sprawą pamięci stałej typu ROM (angielskie Read Only Memory — pamięć, z której można jedynie czytać). Jak sugeruje nazwa, nie mogę ingerować w treść tej pamięci. Została ona raz na zawsze zapisana podczas produkcji i na tym koniec. W tej pamięci zapisany jest mój system operacyjny i język BASIC.

Do zapamiętania programu i następnie do jego wykonania potrzebuję jednak takiej pamięci, którą mógłbym sam zapisywać. W tym celu konstruktorzy wyposażyli mnie w pamięć zapisywalną typu RAM (ang. Random Access Memory — pamięć o swobodnym dostępie). W niej właśnie przechowuję program wpisany z klawiatury, bądź z innego urządzenia zewnętrznego np. magnetofonu. W niej także umieszczam wszystkie liczby i teksty, z których korzystam podczas realizacji programu.

Ten rodzaj pamięci ma jednak bardzo nieprzyjemną cechę. Po odłączeniu zasilania, zapominam wszystko, co było zawarte w RAM. Dlatego za każdym razem, po włączeniu komputera trzeba od nowa wczytywać program.

Niedogodność ta spędza sen z oczu konstruktorom. Niektórzy moi koledzy posiadają akumulatorowe zasilanie, podtrzymujące pamięć RAM po wyłączeniu zasilania. Są prowadzone także próby z innymi rodzajami pamięci, np. pamięć pęcherzykowa. Posiada ona wszystkie zalety RAM, a dodatkowo nie kasuje się po wyłączeniu komputera. To jednak będzie już dotyczyło moich wnuków, choć — jak już kiedyś powiedziałem — my, komputery, stajemy się dziadkami bardzo szybko.

Twój Komputer



BIZNES MEN

„... Trochę się też bałem podróżując pierwszy raz pociągiem z Lyonu do Paryża, ale w końcu wszystko dobrze poszło..."

Czternastoletni uczeń szkoły podstawowej, Cyryl de Vignemont, zaczął swoją „karierę" bardzo wcześnie, bo w wieku 10 lat. Będąc słabego zdrowia nie mógł, jak inni chłopcy w jego wieku, uczęszczać regularnie do szkoły. Uczył się więc na kursach korespondencyjnych, które pozostawiały mu dużo wolnego czasu. Nie marnował go, wyżywał się w programowaniu swojego ZX-80. Długo męczył rodziców o Apple 2, na którym chciał napisać program użytkowy, coś w rodzaju bazy danych. „To niezbyt skomplikowane i o wiele bardziej zabawne niż gry" — mówi.

Program powstał dość szybko i Cyryl zdecydował przedstawić go wydawcom. Bałem się bardzo, że zostanę odesłany. Miałem wtedy nieco ponad jedenaście lat. Dzwoniłem do różnych firm. Usiłowałem mówić poprawnie, pogrubiałem głos. Nareszcie udało mi się ustalić spotkanie, już nie mogli mnie spławić przez telefon. Po czym dodaje jeszcze: Trochę się też bałem podróżując pierwszy raz pociągiem z Lyonu do Paryża, ale w końcu wszystko dobrze poszło.

Pewny siebie i swojego programu Cyryl wkroczy do biura jednego z producentów. Gdy zobaczyliśmy — mówi kierownik firmy — w pierwszej chwili nie mogliśmy w to uwierzyć. Najciekawsze, że program rzeczywiście był dobry.

O chłopcu i jego bazie danych zaczynało być coraz głośniej. Wkrótce program trafił na rynek, a na bankowym koncie Cyryla znalazło się — jak sam mówi — „kilka zer po przecinku". Dwa nowe komputery Liza i MacIntosh, będące częścią wynagrodzenia pozwolą Cyrylowi na pisanie nowych programów.

Tymczasem okazuje się, że program jest wysoko oceniany przez profesjonalistów. Ci ostatni nie wiedzą oczywiście, że został on stworzony przez kilkunastoletniego chłopca. I lepiej, żeby o tym nie wiedzieli: gdyby historia miała miejsce w USA — wszyscy byliby dumni z Cyryla. We Francji jednak nie ufano by mu.

Dlatego Cyryl pracuje cały czas anonimowo: pisze teraz swój drugi duży program na MacIntosha. Tym razem będzie to test sprawdzający wiadomości z jakiejś dziedziny, np. geografii.

Co dalej? Chciałbym stworzyć własną firmę komputerową, zajmującą się pisaniem programów — mówi młody programista. Powiecie: ma jeszcze czas. Tak, mam czas, ale wolałbym mimo wszystko, aby nastąpiło to jeszcze przed Gwiazdką.

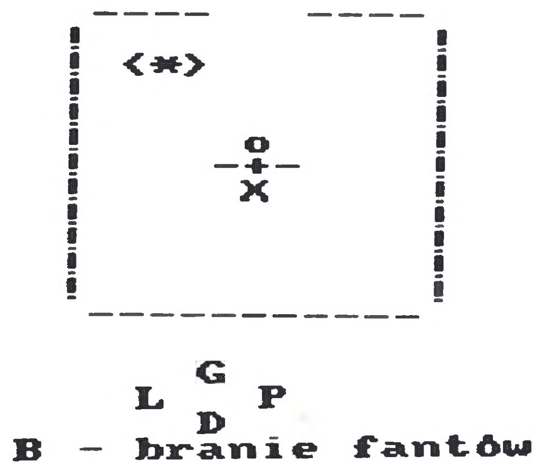
No cóż, może mu się uda.

(JZ)

SKARB KUBUSIA

Widzieliście już z pewnością wiele gier opartych na takiej zasadzie. Tym razem jednak sami spróbujmy napisać podobną. Tak jak to robiliśmy w poprzednim programie — „Spadochroniarze", nie będziemy korzystać z funkcji graficznych, różnych dla każdego mikrokomputera. Aby nasz program był uniwersalny zastosujmy tylko instrukcje PRINT.

Fragment programu od linii 530 do 680 to właśnie rysowanie — za pomocą instrukcji PRINT — jednej komnaty wraz z Kubusiem i ewentualnie ze skarbem czy fantami (rys. 1). Komnaty różnią się mię-



Rys. 1

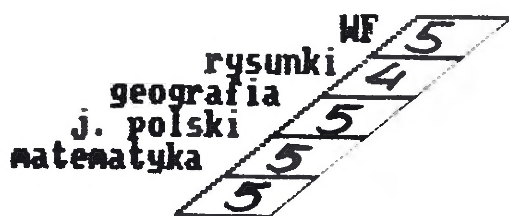
dzy sobą jedynie rozmieszczeniem wejść: gdrzwi, ddrzwi, ldrzwi, pdrzwi. Pozostaje pytanie jak nauczyć nasz komputer — w możliwie najprostszy i najkrótszy sposób — planu całego zamku. Najpierw jednak poznamy nowe pojęcie — tablica, okaże się nam bardzo przydatne.

Wyobraźmy sobie, że dla porządku postanowiliśmy wszystkie nasze notatki przechowywać na półkach (po jednej informacji na półkę). Jeśli więc teraz chcemy zapamiętać jedną rzecz np. ocenę z matematyki wystarczy nam jedna półka (rys. 2a). Jeśli jednak chcemy zanotować



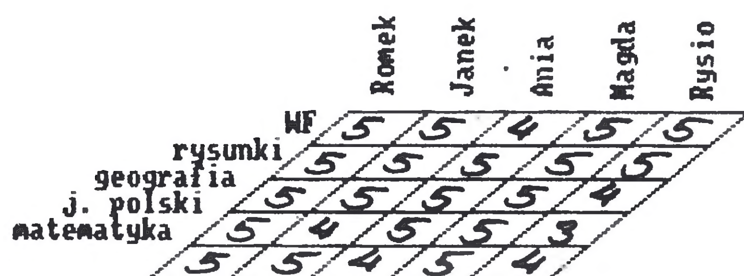
Rys. 2a

oceny z kilku przedmiotów wygodnie będzie skorzystać z kilku półek ustawionych obok siebie (rys. 2b). Sprawa nie komplikuje gdy postawimy umieścić tam także oceny kolegów. Oczywiście moglibyśmy po prostu przedłużyć naszą pół-



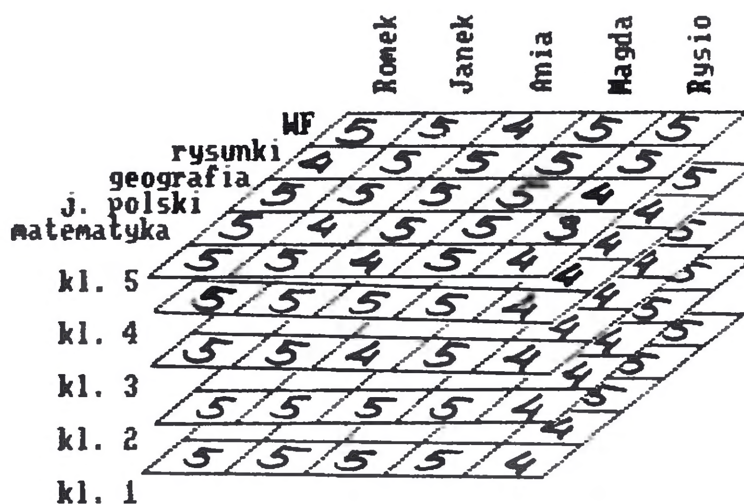
Rys. 2b

kę, lecz byłoby trudno — na pierwszy rzut oka — domyślić się, która z ocen dotyczy którego kolegi i jakiego przedmiotu. Możemy jednak ustawić półki nieco inaczej, w kilku rzędach (rys. 2c). W



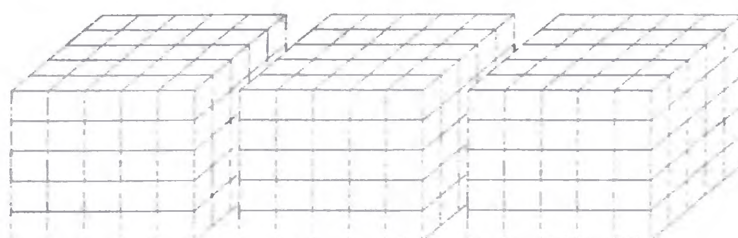
Rys. 2c

pierwszym rzędzie umieścimy nasze oceny, a w następnych, kolegów. Nasze półki możemy również ustawić piętrowo. Każde piętro może np. zawierać oceny na świadectwach z kolejnych klas (rys. 2d).



Rys. 2d

Przygotowując informacje dla komputera często korzystamy z tej samej zasady. Nasze półki nazywamy w tym przypadku tablicami. Rysunek 2b ilustruje tablice jednowymiarową, rys. 2c — dwuwymiarową, natomiast rys. 2d — trójwymiarową. Oczywiście to jeszcze nie koniec. Możemy tworzyć tablice większej liczbie wymiarów ale wówczas trudniej przedstawić ich graficzną ilustrację. Tablice o czterech wymiarach możemy wyobrazić sobie tak jak to zostało przedstawione na rys. 2e (czwarty wymiar to numer kostki).



Rys. 2e

W naszym programie określimy sobie trzy dwuwymiarowe tablice (linie 20-40). W pierwszej zapiszemy czy jest wyjście z danej komnaty (o współrzędnych x,y) prowadzące do góry, w drugiej to samo tylko dla wyjścia w lewo. Trzecia będzie zawierała informacje o skarbach. Sposób określania przejść wyjaśniony jest na rys. 3. Po prostu: „1" — przejście otwarte, „0" — brak przejścia. Odpowiednie wartości zapisujemy w liniach 5000-5090 za instrukcją DATA. Będą one odczytane za pomocą instrukcji READ i wprowadzone do tablic zaraz po uruchomieniu programu (linie 100-160).

```
     0           0           0
0        1   0           1
-----0-------1-------1--------
0        1   0           1
-----1-------1-------0--------
```

Rys. 3

Niektórych z Was pewnie trochę zdziwiło, że zanotowaliśmy tylko górne i lewe przejścia, ale inni już się domyślili o co chodzi. Przecież prawe drzwi jakiejś komnaty są równocześnie lewymi drzwiami komnaty następnej. To samo dotyczy drzwi dolnych. Nie musimy więc zapisywać dwa razy tego samego, wystarczy tylko „zaglądnąć" do sąsiednich komnat (linie 730-740).

Teraz już chyba sami bez trudu odkryjecie resztę tajemnic tego programu. Zwróćcie uwagę na to, skąd komputer wie gdzie znajdują się skarby (i jakie!). Radzę Wam najpierw poznać zasadę działania programu a później dopiero przystąpić do wpisywania go, gdyż w innym razie najdrobniejsza pomyłka może stać się problemem nie do przebycia.

Romek

Rys. 1 Kubuś Literka w komnacie ze skarbem.
Rys. 2a, b, c, d, e Tablice, czyli półki do przechowywania notatek.
Rys. 3 Sposób określania przejść w labiryncie.

```
10 CLS
20 DIM g(11,11) '* deklaracja
30 DIM l(11,11) '*   tablic
40 DIM s(11,11) '*
100  FOR y=1 TO 10 '* petle
110   FOR x=1 TO 10'* zapisu
120   READ g,l      '* tablicy
130   LET g(x,y)=g '* opisu-
140   LET l(x,y)=l '* jacej
150   NEXT x        '* komnaty
160  NEXT y         '*
170  FOR i=1 TO 4 '* petla
180  READ m,n,s   '* zapisu
190  LET s(m,n)=s '* miejsca
200  NEXT i        '* fantow
300 LET x=1     '*  dane
310 LET y=1     '* startowe
320 GOSUB 2000 '*
399 '*** okresl. komnaty ***
400 IF g(x,y)=1 THEN LET gdrzwi$="   "
410 IF l(x,y)=1 THEN LET ldrzwi$=" "
420 IF g(x,y+1)=1 THEN LET ddrzwi$="   "
430 IF l(x+1,y)=1 THEN LET pdrzwi$=" "
440 IF s(x,y)=1 THEN LET skarby$=" * "
450 IF s(x,y)=2 THEN LET skarby$="<*>"
500 CLS '** rys. komnaty **
510  FOR t=1 TO 500
520  NEXT t
530 PRINT " ----";gdrzwi$;"---- "
540 PRINT "!           !       G"
550 PRINT "! ";skarby$;"       !       L   P"
560 PRINT "!           !       D"
570 PRINT "!           !  B - branie fantow"
580 PRINT ldrzwi$;"     o     ";pdrzwi$
590 PRINT ldrzwi$;"    -+-    ";pdrzwi$;"      ";
600  FOR k=1 TO moc
610  PRINT "*";
620  NEXT k
630 PRINT
640 PRINT ldrzwi$;"     X     ";pdrzwi$
650  FOR i=1 TO 4
660  PRINT "!           !"
670  NEXT i
680 PRINT " ----";ddrzwi$;"---- "
700 LET k$=INKEY$ '** klawisz
710 IF k$="g" AND gdrzwi$="   " THEN LET y=y-1
720 IF k$="d" AND ddrzwi$="   " THEN LET y=y+1
730 IF k$="l" AND ldrzwi$=" " THEN LET x=x-1
740 IF k$="p" AND pdrzwi$=" " THEN LET x=x+1
750 IF k$="b" AND skarby$<>"   " THEN GOSUB 1000
760 IF k$="" THEN GOTO 700
900 GOSUB 2000 '** podprog.**
910 GOTO 400
999 '**** branie fantow ****
1000 IF skarby$=" * " THEN LET moc=moc+1: LET s(x,
y)=0
1010 IF skarby$="<*>" AND moc=3 THEN PRINT:PRINT "
Wygrales":STOP
2000 LET gdrzwi$="---"'* ust.
2010 LET ddrzwi$="---"'* par.
2020 LET pdrzwi$="!"  '* kom-
2030 LET ldrzwi$="!"  '* naty
2040 LET skarby$="   "'*
2050 RETURN          '*
4999 '***** przejscia *****
5000 DATA 0,0,0,1,0,0,0,1,0,1,0,1,0,1,0,1,0,1,0,1
5010 DATA 0,0,1,1,1,0,1,1,0,0,0,1,0,1,0,1,1,0,1,1
5020 DATA 1,0,1,1,1,1,0,1,1,1,1,0,1,1,1,0,0,1,1,0
5030 DATA 0,0,0,1,0,1,0,1,0,1,0,0,0,1,1,1,1,0,1,1
5040 DATA 1,0,1,1,1,0,1,1,1,0,1,1,0,1,1,1,1,0,1,1
5050 DATA 1,0,0,0,1,1,1,1,0,0,0,0,0,1,0,1,1,1,1,1
5060 DATA 1,0,0,1,0,1,0,1,1,1,1,0,1,0,0,0,0,1,1,0
5070 DATA 1,0,1,0,0,0,1,1,0,0,1,1,1,1,1,0,1,0,1,0
5080 DATA 0,0,0,1,1,0,0,0,1,1,0,0,0,1,1,1,0,1,0,1
5090 DATA 1,0,0,1,1,1,0,1,0,1,1,1,0,1,0,1,0,1,0,1
5999 '****** fanty ******
6000 DATA 2,6,1,9,3,1,4,9,1,7,8,2
```


KLUCZ DO ENERGII JUTRA

Dokończenie ze str. 32

Czym jest synteza termojądrowa i jak próbuje się ją urzeczywistnić?

Jeszcze obecnie podstawową metodą wytwarzania energii jest spalanie paliwa organicznego. W tym procesie chemicznym atom węgla łączy się z cząsteczką tlenu, w wyniku czego powstaje molekuła dwutlenku węgla, energia zaś wyzwala się w postaci ciepła, które ludzie wykorzystują już od wielu tysięcy lat.

Reakcje syntezy jądrowej są bardzo podobne do chemicznej reakcji spalania, z tym, że zamiast atomów łączą się jądra. Przypomnijmy w tym miejscu, że jądra atomowe składają się z protonów i neutronów, a w zasadzie każdy pierwiastek chemiczny może mieć jądra z różną liczbą neutronów. Takie odmiany pierwiastków nazywają się izotopami.

Najprostszy pierwiastek chemiczny — wodór, posiada trzy izotopy. Jądrem zwykłego wodoru jest proton, jądro deuteru (D) składa się z protonu i neutronu, a jądro trytu (T) — z protonu i dwóch neutronów.

Synteza termojądrowa polega właśnie na połączeniu się obu izotopów wodoru — deuteru i trytu. Tworzy się z nich jądro helu oraz jeden neutron, które rozlatują się z miejsca reakcji z olbrzymimi prędkościami (np. prędkość neutronu wynosi ok. 50 000 km/s). Dodajmy, że w wyniku połączenia się pojedynczych izotopów deuteru i trytu powstaje 4 miliony razy więcej energii, niż podczas utlenienia się (czyli spalenia) jednego atomu węgla!

Wiadomo, że aby mogła się rozpocząć chemiczna reakcja spalania, należy zmusić atomy, aby się ze sobą zderzyły, a w tym celu niezbędna jest wstępnie dodatkowa porcja energii w formie „podpałki". Zainicjowanie reakcji syntezy jądrowej również wymaga nagrzania mieszaniny deuteru i trytu. Stąd też reakcje syntezy jądrowej nazywane są termojądrowymi. Na przykład aby „podpalić" najłatwiej „zapalającą się" mieszaninę deuteru i trytu temperatura „zapałki" powinna przekraczać 50 mln stopni. W bombie wodorowej jako zapalnik wykorzystuje się bombę atomową. **Jaką „zapałkę" chcą uczeni zastosować, jak podgrzewać mieszaninę deuteru i trytu, aby uzyskać kontrolowaną syntezę termojądrową?** Zdaniem wybitnego uczonego radzieckiego akademika **Nikołaja Basowa** najlepiej tę funkcję mogą spełniać lasery.

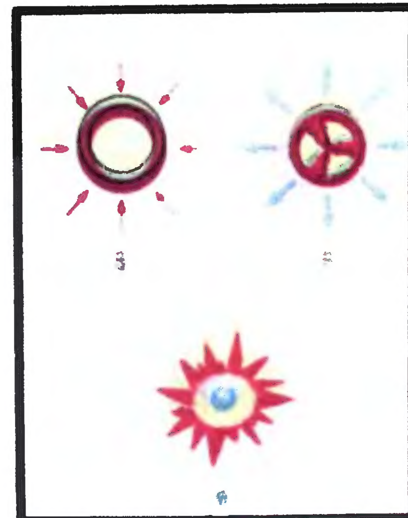
Nikołaj Basow jest tym właśnie uczonym, który za wynalezienie laserów otrzymał w 1964 roku Nagrodę Nobla. Obecnie kierowany przez niego Instytut Fizyki Akademii Nauk ZSRR jest wiodącą na świecie placówką w zakresie badań nad **laserową syntezą termojądrową**.

Będąc w lutym br. u profesora Basowa, w Moskwie, miałem okazję zapoznania się z pracą eksperymentalnych urządzeń noszących efektowne nazwy „Kalmar", „Delfin-1" i „Delfin-2", w których bada się reakcje syntezy jądrowej.

Oto schemat laserowej syntezy termojądrowej. Tarcza termojądrowa — czyli malutka szklana kula, wypełniona mieszaniną deuteru i trytu — napromieniowana jest ze wszystkich stron zogniskowanymi na niej wiązkami promieniowania laserowego. Wskutek potężnej energii promieniowania powłoka tarczy zaczyna parować. Parowanie to jest tak dalece intensywne, że siła odrzutu będzie mocno sprężać pozostałą część kulki w kierunku jej środka. Obliczenia wykazują, że gęstość materii może wówczas osiągnąć wartość tysięcy gramów na centymetr sześcienny! Temperatura w środku ściśniętej wiązkami laserowymi tarczy podniesie się do kilkuset milionów stopni, a wówczas rozpocznie się reakcja syntezy termojądrowej. Reakcja ta ustanie z chwilą, gdy dojdzie do mikrowybuchu, w rezultacie którego plazma rozpadnie się.

Zapytany o stan badań nad laserową syntezą termojądrową, profesor Basow uśmiecha się i mówi: — *Do dnia dzisiejszego przeszliśmy już bardzo długą drogę w tym kierunku i nie napotkaliśmy na żadne przeciwwskazania. Więcej, cała wiedza, jaką zdobyliśmy daje nam przekonanie, praktycznie stuprocentowe, że w oparciu o metodę laserową można stworzyć bardzo dobry reaktor termojądrowy.*

Schemat laserowej syntezy termojądrowej. Promienie laserów równomiernie nagrzewają kulkę zawierającą mieszaninę deuteru i trytu (a). Nagrzewanie to przebiega tak szybko, że środek kulki pozostaje chłodny, a jej powłoka wybuchowo wyparowuje (b). Powstałe podczas tego parowania siły odrzutu sprężają plazmę i nagrzewają ją do temperatury niezbędnej do rozpoczęcia się reakcji syntezy termojądrowej (c).



Jak będzie zbudowany przyszły reaktor termojądrowy?

Jeden z projektów przewiduje, że będzie on miał postać kulistej komory o średnicy około 10 m. Specjalne urządzenia rytmicznie podawać będą na sam środek komory kulki paliwa termojądrowego. Przez przezroczyste okna w ścianach komory kulki te będą napromieniowywane za pomocą laserów. Po każdym wybuchu termojądrowym, zanim nastąpi kolejna „salwa" laserowa, potężne pompy wytwarzać będą próżnię w komorze detonacyjnej. Aby uchronić ściany komory przed rozerwaniem przez neutrony i jądra helu, skonstruowano ją z porowatego materiału, którego powierzchnia przed wybuchem pokrywana jest warstwą płynnego litu (tzw. mokra ściana). Lit dobrze pochłania produkty syntezy i chroni ścianę komory przed ich zbyt intensywnym oddziaływaniem. Za pierwszą ścianą komory znajduje się druga, zaś przestrzeń między nimi też wypełnia płynny lit, pełniący rolę nośnika ciepła wykorzystywanego następnie do uzyskiwania pary i napędzania turbin elektrowni.

Trudno na razie jeszcze powiedzieć, kiedy powyższy projekt zostanie zrealizowany. Ale warto poznać opinię akademika **Jewgienija Wielichowa**, szefa programu termojądrowego ZSRR, który uważa: *„Istnieją wszelkie podstawy do twierdzenia, że badania w dziedzinie kontrolowanej syntezy termojądrowej znajdują się obecnie na etapie szybkiego rozwoju. Z trzech parametrów koniecznych do przebiegu reakcji syntezy deuteru i trytu (temperatura, gęstość plazmy i czas jej utrzymania) każdy z nich z osobna lub w parze z jednym z dwóch pozostałych został osiągnięty. Obecne zadanie polega na otrzymaniu niezbędnych wartości wszystkich trzech parametrów w jednym eksperymencie. Sądząc po tempie, z jakim rozwijają się te badania, cel ten będzie prawdopodobnie osiągnięty w ciągu kilku najbliższych lat".*

Warto podkreślić, że dzięki opanowaniu umiejętności sterowania syntezą termojądrową ludzkość nie tylko uzyska dostęp do niewyczerpalnego w perspektywie źródła energii. Otóż reaktor termojądrowy, w odróżnieniu od jądrowego, nie produkuje materiałów rozszczepialnych, które stanowią surowiec do budowy broni jądrowej i termojądrowej.

Waldemar Siwiński

Są takie idee i projekty naukowe, które w pewnych okresach czasu ogniskują uwagę i wysiłek najwybitniejszych uczonych świata. Do takich wyzwań stojących przed światową nauką należy obecnie kontrolowana synteza termojądrowa, która może zapewnić ludzkości dostatek energii na setki lat.

Dokończenie na str. 31



KLUCZ DO ENERGII JUTRA